Packaging research artefacts with RO-Crate

Stian Soiland-Reyes [a,b,*], Peter Sefton [c], Mercè Crosas [d], Leyla Jael Castro [e], Frederik Coppens [f], José M. Fernández [g], Daniel Garijo [h], Björn Grüning [i], Marco La Rosa [j], Simone Leo [k], Eoghan Ó Carragáin [l], Marc Portier [m], Ana Trisovic [n], RO-Crate Community [o], Paul Groth [p], Carole Goble [q]

[a] *Department of Computer Science, The University of Manchester, UK*
E-mail: soiland-reyes@manchester.ac.uk; ORCID: https://orcid.org/0000-0001-9842-9718
[b] *Informatics Institute, University of Amsterdam, The Netherlands*
[c] *Faculty of Science, University Technology Sydney, Australia*
E-mail: Peter.Sefton@uts.edu.au; ORCID: https://orcid.org/0000-0002-3545-944X
[d] *Institute for Quantitative Social Science, Harvard University, Cambridge, MA, USA*
ORCID: https://orcid.org/0000-0003-1304-1939
[e] *ZB MED Information Centre for Life Sciences, Cologne, Germany*
ORCID: https://orcid.org/0000-0003-3986-0510
[f] *VIB-UGent Center for Plant Systems Biology, Gent, Belgium*
ORCID: https://orcid.org/0000-0001-6565-5145
[g] *Barcelona Supercomputing Center, Barcelona, Spain*
ORCID: https://orcid.org/0000-0002-4806-5140
[h] *Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain*
ORCID: https://orcid.org/0000-0003-0454-7145
[i] *Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg, Freiburg, Germany*
ORCID: https://orcid.org/0000-0002-3079-6586
[j] *PARADISEC, Melbourne, Australia*
ORCID: https://orcid.org/0000-0001-5383-6993
[k] *Center for Advanced Studies, Research, and Development in Sardinia (CRS4), Pula (CA), Italy*
ORCID: https://orcid.org/0000-0001-8271-5429
[l] *University College Cork, Ireland*
ORCID: https://orcid.org/0000-0001-8131-2150
[m] *Vlaams Instituut voor de Zee, Oostende, Belgium*
ORCID: https://orcid.org/0000-0002-9648-6484
[n] *Institute for Quantitative Social Science, Harvard University, Cambridge, MA, USA*
ORCID: https://orcid.org/0000-0003-1991-0533
[o] *https://www.researchobject.org/ro-crate/community (see appendix B)*
[p] *Informatics Institute, University of Amsterdam, The Netherlands*
E-mail: p.t.groth@uva.nl; ORCID: https://orcid.org/0000-0003-0183-6910
[q] *Department of Computer Science, The University of Manchester, UK*

E-mail: carole.goble@manchester.ac.uk; ORCID: https://orcid.org/0000-0003-1219-2137

Abstract. An increasing number of researchers support reproducibility by including pointers to and descriptions of datasets, software and methods in their publications. However, scientific articles may be ambiguous, incomplete and difficult to process by automated systems. In this paper we introduce RO-Crate, an open, community-driven, and lightweight approach to packaging research artefacts along with their metadata in a machine readable manner. RO-Crate is based on Schema.org annotations in JSON-LD, aiming to establish best practices to formally describe metadata in an accessible and practical way for their use in a wide variety of situations.

An RO-Crate is a structured archive of all the items that contributed to a research outcome, including their identifiers, provenance, relations and annotations. As a general purpose packaging approach for data and their metadata, RO-Crate is used across multiple areas, including bioinformatics, digital humanities and regulatory sciences. By applying ”just enough” Linked Data standards, RO-Crate simplifies the process of making research outputs FAIR while also enhancing research reproducibility.

An RO-Crate for this article[1] is archived at `https://doi.org/10.5281/zenodo.5146227`

Keywords: Data publishing, Data packaging, FAIR, Linked Data, Metadata, Reproducibility, Research Object

1. Introduction

The move towards Open Science has increased the need and demand for the publication of artefacts of the research process [1]. This is particularly apparent in domains that rely on computational experiments; for example, the publication of software, datasets and records of the dependencies that such experiments rely on [2].

It is often argued that the publication of these assets, and specifically software [3], workflows [4] and data, should follow the FAIR principles [5]; namely, that they are Findable, Accessible, Interoperable and Reusable. These principles are agnostic to the *implementation* strategy needed to comply with them. Hence, there has been an increasing amount of work in the development of platforms and specifications that aim to fulfil these goals [6].

Important examples include data publication with rich metadata (e.g. Zenodo [7]), domain-specific data deposition (e.g. PDB [8]) and following practices for reproducible research software [9] (e.g. use of containers). While these platforms are useful, experience has shown that it is important to put greater emphasis on the interconnection of the multiple artefacts that make up the research process [10].

The notion of **Research Objects** [11] (RO) was introduced to address this connectivity, providing semantically rich *aggregations* of (potentially distributed) resources with a layer of structure over a research study; this is then to be delivered in a *machine-readable format*.

A Research Object combines the ability to bundle multiple types of artefacts together, such as spreadsheets, code, examples, and figures. The RO is augmented with annotations and relationships that describe the artefacts' *context* (e.g. a CSV being used by a script, a figure being a result of a workflow).

This notion of ROs provides a compelling vision as an approach for implementing FAIR data. However, existing Research Object implementations require a large technology stack [12], are typically tailored to a particular platform and are also not easily usable by end-users.

[*]Corresponding author. E-mail: soiland-reyes@manchester.ac.uk.

[1]<https://www.researchobject.org/2021-packaging-research-artefacts-with-ro-crate/>

To address this gap, a new community came together [13] to develop **RO-Crate** — an *approach to package and aggregate research artefacts with their metadata and relationships*. The aim of this paper is to introduce RO-Crate and assess it as a strategy for making multiple types of research artefacts FAIR. Specifically, the contributions of this paper are as follows:

1. An introduction to RO-Crate, its purpose and context;
2. A guide to the RO-Crate community and tooling;
3. Examples of RO-Crate usage, demonstrating its value as connective tissue for different artefacts from different communities.

The rest of this paper is organised as follows. We first describe RO-Crate through its development methodology that formed the RO-Crate concept, showing its foundations in Linked Data and emerging principles. We then define RO-Crate technically, before we introduce the community and tooling. We move to analyse RO-Crate with respect to usage in a diverse set of domains. Finally, we present related work and conclude with some remarks including RO-Crate highlights and future work. The appendix adds a formal definition of RO-Crate using First-Order logic.

2. RO-Crate

RO-Crate aims to provide an approach to packaging research artefacts with their metadata that can be easily adopted. To illustrate this, let us imagine a research paper reporting on the sequence analysis of proteins obtained from an experiment on mice. The sequence output files, sequence analysis code, resulting data and reports summarising statistical measures are all important and inter-related research artefacts, and consequently would ideally all be co-located in a directory and accompanied with their corresponding metadata. In reality, some of the artefacts (e.g. data or software) will be recorded as external reference to repositories that are not necessarily following the FAIR principles. This conceptual directory, along with the relationships between its constituent digital artefacts, is what the RO-Crate model aims to represent, linking together all the elements of an experiment that are required for the experiment's reproducibility and reusability.

The question then arises as to how the directory with all this material should be packaged in a manner that is accessible and usable by others. This means programmatically and automatically accessible by machines and human readable. A de facto approach to sharing collections of resources is through compressed archives (e.g. a zip file). This solves the problem of "packaging", but it does not guarantee downstream access to all artefacts in a programmatic fashion, nor describe the role of each file in that particular research. Both features, the ability to automatically access and reason about an object, are crucial and lead to the need for explicit metadata about the contents of the folder, describing each and linking them together.

Examples of metadata descriptions across a wide range of domains[2] abound within the literature, both in research data management [14–16] and within library and information systems[3] [17, 18]. However, many of these approaches require knowledge of metadata schemas, particular annotation systems, or the use of complex software stacks. Indeed, particularly within research, these requirements have led to a lack of adoption and growing frustration with current tooling and specifications [19–21].

RO-Crate seeks to address this complexity by:

[2]<https://rdamsc.bath.ac.uk/scheme-index>
[3]<https://www.loc.gov/librarians/standards>

1. being conceptually simple and easy to understand for developers;
2. providing strong, easy tooling for integration into community projects;
3. providing a strong and opinionated guide regarding current best practices;
4. adopting de-facto standards that are widely used on the Web.

In the following sections we demonstrate how the RO-Crate specification and ecosystem achieve these goals.

2.1. Development Methodology

It is a good question as to what base level we assume for 'conceptually simple'. We take simplicity to apply at two levels: for the *developers* who produce the platforms and for the *data practitioners* and users of those platforms.

For our development methodology we followed the mantra of working closely with a small group to really get a deep understanding of requirements and ensure rapid feedback loops. We created a pool of early adopter projects from a range of disciplines and groups, primarily addressing developers of platforms. Thus the base level for simplicity was **developer friendliness**.

We assumed a developer familiar with making Web applications with JSON data (who would then learn how to make *RO-Crate JSON-LD*), which informed core design choices for our JSON-level documentation approach and RO-Crate serialization (section 2.3). Our group of early adopters, growing as the community evolved, drove the RO-Crate requirements and provided feedback through our multiple communication channels including bi-monthly meetings, which we describe in section 2.4 along with the established norms.

Addressing the simplicity of understanding and engaging with RO-Crate by data practitioners is through the platforms, for example with interactive tools (section 3) like Describo[4] [22] and Jupyter notebooks [23], and by close discussions with domain scientists on how to appropriately capture what they determine to be relevant metadata. This ultimately requires a new type of awareness and learning material separate from developer specifications, focusing on the simplicity of extensibility to serve the user needs, along with user-driven development of new RO-Crate Profiles specific for their needs (section 4).

2.2. Conceptual Definition

A key premise of RO-Crate is the existence of a wide variety of resources on the Web that can help describe research. As such, RO-Crate relies on the Linked Data principles [24]. Figure 1 shows the main conceptual elements involved in an RO-Crate: The RO-Crate Metadata File (top) describes the Research Object using structured metadata including external references, coupled with the contained artefacts (bottom) bundled and described by the RO-Crate.

The conceptual notion of a *Research Object* [11] is thus realized with the RO-Crate model and serialized using Linked Data constructs within the RO-Crate metadata file.

2.2.1. Linked Data as a foundation

The **Linked Data** principles [29] (use of IRIs[5] to identify resources (i.e. artefacts), resolvable via HTTP, enriched with metadata and linked to each other) are core to RO-Crate; therefore IRIs are

[4]<https://uts-eresearch.github.io/describo/>

[5]**IRI**s [30] are a generalisation of *URI*s (which include well-known http/https URLs), permitting international Unicode characters without '

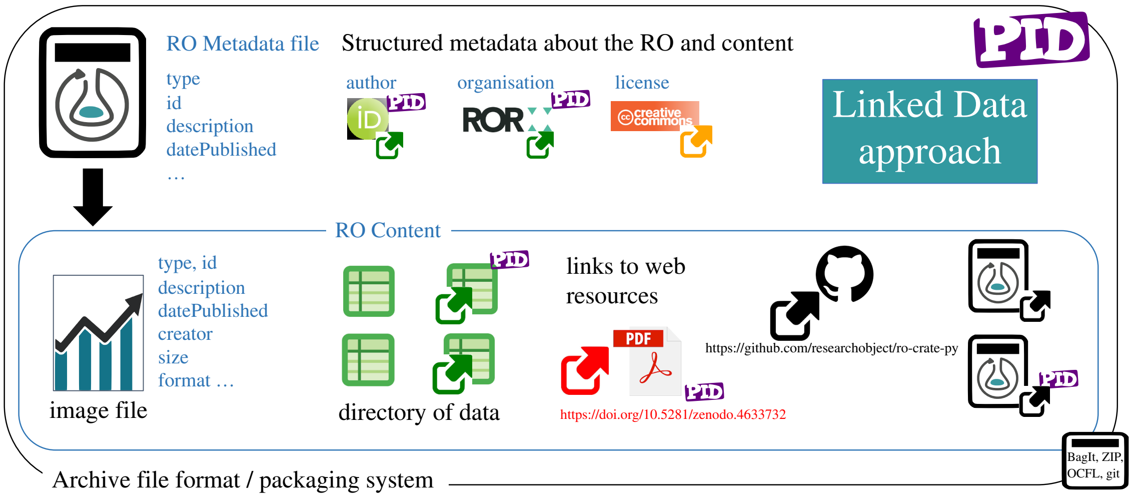

Fig. 1. **Conceptual overview of RO-Crate**. A *Persistent Identifier* (PID) [25] points to a *Research Object* (RO), which may be archived using different packaging approaches like BagIt [26], OCFL [27], git or ZIP. The RO is described within a *RO-Crate Metadata File*, providing identifiers for *authors* using ORCID, *organisations* using Research Organization Registry (ROR) [28] and licences such as Creative Commons using SPDX identifiers. The *RO-Crate content* is further described with additional metadata following a Linked Data approach. Data can be embedded files and directories, as well as links to external Web resources, PIDs and nested RO-Crates.

used to identify an RO-Crate, its constituent parts and metadata descriptions, and the properties and classes used in the metadata.

RO-Crates are *self-described* and follow the Linked Data principles to describe all of their resources in both human and machine readable manner. Hence, resources are identified using global identifiers (absolute IRIs) where possible; and relationships between two resources are defined with links.

The foundation of Linked Data and shared vocabularies also means that multiple RO-Crates and other Linked Data resources can be indexed, combined, queried, validated or transformed using existing Semantic Web technologies such as SPARQL[6], SHACL[7] and well established *knowledge graph* triple stores like Apache Jena[8] and OntoText GraphDB[9].

The possibilities of consuming[10] RO-Crate metadata with such powerful tools gives another strong reason for using Linked Data as a foundation. This use of mature Web[12] technologies also means its developers and consumers are not restricted to the Research Object aspects that have already been specified by the RO-Crate community, but can extend and integrate RO-Crate in multiple standardized ways.

[6]<https://www.w3.org/TR/sparql11-overview>
[7]<https://www.w3.org/TR/shacl/>
[8]<https://jena.apache.org/>
[9]<https://www.ontotext.com/products/graphdb/>
[10]Some consideration is needed in processing of RO-Crates as knowledge graphs, e.g. establishing absolute URIs for files inside a ZIP archive, detailed in the RO-Crate specification: https://www.researchobject.org/ro-crate/1.1/appendix/relative-uris.html[11]
[12]Note that an RO-Crate is not required to be published on the Web, see section 2.2.2.

2.2.2. RO-Crate is a self-described container

An RO-Crate is defined[13] as a self-described **Root Data Entity** that describes and contains *data entities*, which are further described by referencing *contextual entities*. A **data entity** is either a *file* (i.e. a byte sequence stored on disk somewhere) or a *directory* (i.e. set of named files and other directories). A file does not need to be stored inside the RO-Crate root, it can be referenced via a PID/IRI. A **contextual entity** exists outside the information system (e.g. a Person, a workflow language) and is stored solely by its metadata. The representation of a **data entity** as a byte sequence makes it possible to store a variety of research artefacts including not only data but also, for instance, software and text.

The Root Data Entity is a directory, the *RO-Crate Root*, identified by the presence of the **RO-Crate Metadata File** `ro-crate-metadata.json` (top of Figure 1). This is a JSON-LD file that describes the RO-Crate, its content and related metadata using Linked Data. JSON-LD is a W3C standard RDF serialisation that has become popular as it is easy to read by humans while also offers some advantages for data exchange on the Internet. JSON-LD, a subset of the widely supported and well-known JSON format, has tooling available for many programming languages[14].

The minimal requirements for the root data entity metadata[15] are `name`, `description` and `datePublished`, as well as a contextual entity identifying its `license` — additional metadata are commonly added to entities depending on the purpose of the particular RO-Crate.

RO-Crates can be stored, transferred or published in multiple ways, e.g. BagIt [26], Oxford Common File Layout [27] (OCFL), downloadable ZIP archives in Zenodo or through dedicated online repositories, as well as published directly on the Web, e.g. using GitHub Pages[16]. Combined with Linked Data identifiers, this caters for a diverse set of storage and access requirements across different scientific domains, from metagenomics workflows producing hundreds of gigabytes of genome data to cultural heritage records with access restrictions for personally identifiable data. Specific RO-Crate profiles[17] may constrain serialization and publication expectations, and require additional contextual types and properties.

2.2.3. Data Entities are described using Contextual Entities

RO-Crate distinguishes between data and contextual entities[18] in a similar way to HTTP terminology's early attempt to separate *information* (data) and *non-information* (contextual) resources [31]. Data entities are usually files and directories located by relative IRI references within the RO-Crate Root, but they can also be Web resources or restricted data identified with absolute IRIs, including *Persistent Identifiers* (PIDs) [25].

As both types of entities are identified by IRIs, their distinction is allowed to be blurry; data entities can be located anywhere and be complex, while contextual entities can have a Web presence beyond their description inside the RO-Crate. For instance `https://orcid.org/0000-0002-1825-0097` is primarily an identifier for a person, but secondarily it is also a Web page and a way to refer to their academic work.

A particular IRI may appear as a contextual entity in one RO-Crate and as a data entity in another; the distinction lies in the fact that data entities can be considered to be *contained* or captured by

[13]<https://www.researchobject.org/ro-crate/1.1/structure.html#ro-crate-metadata-file-ro-crate-metadatajson>
[14]<https://json-ld.org/#developers>
[15]<https://www.researchobject.org/ro-crate/1.1/root-data-entity.html#direct-properties-of-the-root-data-entity>
[16]<https://pages.github.com/>
[17]<https://www.researchobject.org/ro-crate/1.2-DRAFT/profiles.html>
[18]<https://www.researchobject.org/ro-crate/1.1/contextual-entities.html#contextual-vs-data-entities>

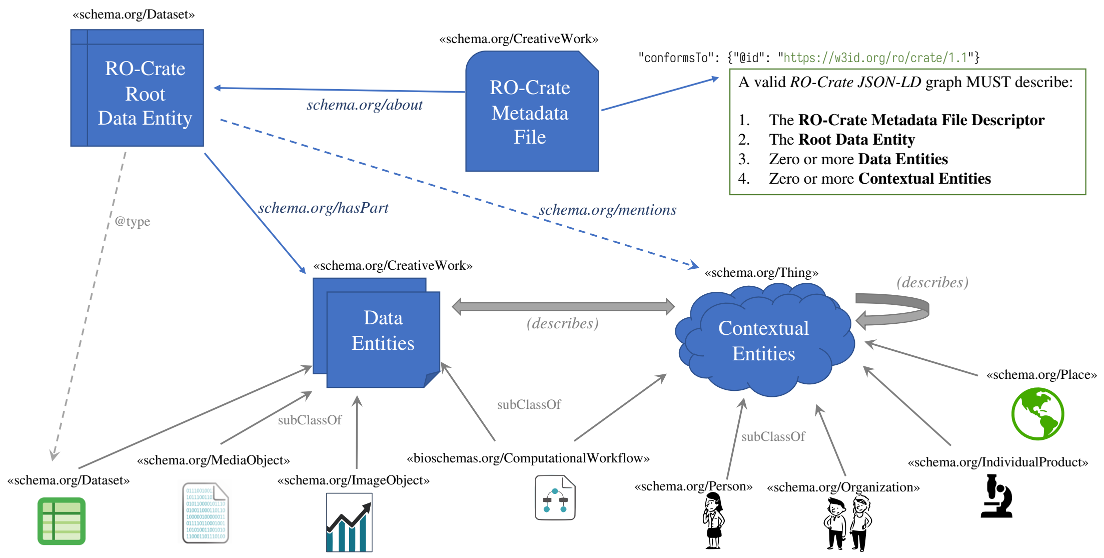
«schema.org/Dataset»
RO-Crate Root Data Entity
«schema.org/CreativeWork»
RO-Crate Metadata File
schema.org/about
"conformsTo": {"@id": "https://w3id.org/ro/crate/1.1"}
A valid RO-Crate JSON-LD graph MUST describe:
1. The RO-Crate Metadata File Descriptor
2. The Root Data Entity
3. Zero or more Data Entities
4. Zero or more Contextual Entities
@type
schema.org/hasPart
schema.org/mentions
«schema.org/CreativeWork»
Data Entities
«schema.org/Thing»
Contextual Entities
(describes)
(describes)
subClassOf
subClassOf
subClassOf
«schema.org/Place»
«schema.org/IndividualProduct»
«schema.org/Dataset»
«schema.org/MediaObject»
«schema.org/ImageObject»
«bioschemas.org/ComputationalWorkflow»
«schema.org/Person»
«schema.org/Organization»

Fig. 2. **Simplified UML class diagram of RO-Crate.** The *RO-Crate Metadata File* conforms to a version of the specification; and contains a JSON-LD graph [32] that describes the entities that make up the RO-Crate. The *RO-Crate Root Data Entity* represent the Research Object as a dataset. The RO-Crate aggregates *data entities* (`hasPart`) which are further described using *contextual entities* (which may include aggregated and non-aggregated data entities). Multiple types and relations from Schema.org allow annotations to be more specific, including figures, nested datasets, computational workflows, people, organisations, instruments and places. Contextual entities not otherwise cross-referenced from other entities' properties (*describes*) can be grouped under the root entity (`mentions`).

that RO-Crate (*RO Content* in Figure 1), while contextual entities mainly *explain* an RO-Crate or its content (although this distinction is not a formal requirement).

In RO-Crate, a referenced contextual entity (e.g. a person identified by ORCID) should always be described within the RO-Crate Metadata File with at least a *type* and *name*, even where their PID might resolve to further Linked Data. This is so that clients are not required to follow every link for presentation purposes, for instance HTML rendering. Similarly any imported extension terms[19] would themselves also have a human-readable description in the case where their PID does not resolve to human-readable documentation.

Figure 2 shows a simplified UML class diagram of RO-Crate, highlighting the different types of data entities and contextual entities that can be aggregated and related. While an RO-Crate would usually contain one or more data entities (`hasPart`), it may also be a pure aggregation of contextual entities (`mentions`).

2.2.4. Guide through Recommended Practices

RO-Crate as a specification aims to build a set of recommended practices on how to practically apply existing standards in a common way to describe research outputs and their provenance, without having to learn each of the underlying technologies in detail.

As such, the RO-Crate 1.1[20] specification [33] can be seen as an opinionated and example-driven guide to writing Schema.org[21] [34] metadata as JSON-LD [32] (see section 2.3), which leaves it open

[19]<https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#extending-ro-crate>
[20]<https://w3id.org/ro/crate/1.1>
[21]<https://schema.org/>

for implementers to include additional metadata using other Schema.org types and properties, or even additional Linked Data vocabularies/ontologies or their own ad-hoc terms.

However the primary purpose of the RO-Crate specification is to assist developers in leveraging Linked Data principles for the focused purpose of describing Research Objects in a structured language, while reducing the steep learning curve otherwise associated in Semantic Web adaptation, like development of ontologies, identifiers, namespaces, and RDF serialization choices.

2.2.5. Ensuring Simplicity

One aim of RO-Crate is to be conceptually simple. This simplicity has been repeatedly checked and confirmed through an informal community review process. For instance, in the discussion on supporting ad-hoc vocabularies[22] in RO-Crate, the community explored potential Linked Data solutions. The conventional wisdom in RDF best practices[23] is to establish a vocabulary with a new URI namespace, formalised using RDF Schema[24] or OWL[25] ontologies. However, this may seem an excessive learning curve for non-experts in semantic knowledge representation, and the RO-Crate community instead agreed on a dual lightweight approach: (i) Document[26] how projects with their own Web-presence can make a pure HTML-based vocabulary, and (ii) provide a community-wide PID namespace under https://w3id.org/ro/terms/[27] that redirect to simple CSV files maintained in GitHub[28].

To further verify this idea of simplicity, we have formalised the RO-Crate definition (see *Appendix A*). An important result of this exercise is that the underlying data structure of RO-Crate, although conceptually a graph, is represented as a depth-limited tree. This formalisation also emphasises the *boundedness* of the structure; namely, the fact that elements are specifically identified as being either semantically *contained* by the RO-Crate as *Data Entities* (`hasPart`) or mainly referenced (`mentions`) and typed as *external* to the Research Object as *Contextual Entities*. It is worth pointing out that this semantic containment can extend beyond the physical containment of files residing within the RO-Crate Root directory on a given storage system, as the RO-Crate data entities may include any data resource globally identifiable using IRIs.

2.2.6. Extensibility and RO-Crate profiles

The RO-Crate specification provides a core set of conventions to describe research outputs using types and properties applicable across scientific domains. However we have found that domain-specific use of RO-Crate will, implicitly or explicitly, form a specialized **profile** of RO-Crate; i.e., *a set of conventions, types and properties that are minimally required and one can expect to be present in that subset of RO-Crates*. For instance, RO-Crates used for exchange of workflows will have to contain a data entity of type `ComputationalWorkflow`, or cultural heritage records should have a `contentLocation`.

Making such profiles explicit allow further reliable programmatic consumption and generation of RO-Crates beyond the core types defined in the RO-Crate specification. Following the RO-Crate mantra of *guidance over strictness*, profiles are mainly *duck-typing* rather than strict syntactic or

[22]<https://github.com/ResearchObject/ro-crate/issues/71>
[23]<https://www.w3.org/TR/swbp-vocab-pub/>
[24]<http://www.w3.org/TR/2014/REC-rdf-schema-20140225/>
[25]<http://www.w3.org/TR/2012/REC-owl2-overview-20121211/>
[26]<https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#adding-new-or-ad-hoc-vocabulary-terms>
[27]<https://w3id.org/ro/terms/>
[28]<https://github.com/ResearchObject/ro-terms>

semantic types, but may also have corresponding machine-readable schemas at multiple levels (file formats, JSON, RDF shapes, RDFS/OWL semantics).

The next version of the RO-Crate specification 1.2 will define a formalization[29] for publishing and declaring conformance to RO-Crate profiles. Such a profile is primarily a human-readable document of before-mentioned expectations and conventions, but may also define a machine-readable profile as a **Profile Crate**: Another RO-Crate that describe the profile and in addition can list schemas for validation, compatible software, applicable repositories, serialization/packaging formats, extension vocabularies, custom JSON-LD contexts and examples (see for example the Workflow RO-Crate profile[30]).

In addition, there are sometimes existing domain-specific metadata formats, but they are either not RDF-based (and thus time-consuming to construct terms for in JSON-LD) or are at a different granularity level that might become overwhelming if represented directly in the RO-Crate Metadata file (e.g. W3C PROV bundle detailing every step execution of a workflow run [35]). RO-Crate allows such *alternative metadata files* to co-exist, and be described as data entities with references to the standards and vocabularies they conform to. This simplifies further programmatic consumption even where no filename or file extension conventions have emerged for those metadata formats.

Section 4 examines the observed specializations of RO-Crate use in several domains and their emerging profiles.

2.3. Technical implementation of the RO-Crate model

The RO-Crate conceptual model has been realised using JSON-LD and Schema.org in a prescriptive form as discussed in section 2.2. These technical choices were made to cater for simplicity from a developer perspective (as introduced in section 2.1).

JSON-LD[31] [32] provides a way to express Linked Data as a JSON structure, where a *context* provides mapping to RDF properties and classes. While JSON-LD cannot map arbitrary JSON structures to RDF, we found that it does lower the barrier compared to other RDF syntaxes, as the JSON syntax nowadays is a common and popular format for data exchange on the Web.

However, JSON-LD alone has too many degrees of freedom and hidden complexities for software developers to reliably produce and consume without specialised expertise or large RDF software frameworks. A large part of the RO-Crate specification is therefore dedicated to describing the acceptable subset of JSON structures.

2.3.1. RO-Crate JSON-LD

RO-Crate mandates[32] the use of flattened, compacted JSON-LD in the RO-Crate Metadata file `ro-crate-metadata.json`[33] where a single `@graph` array contains all the data and contextual entities in a flat list. An example can be seen in the JSON-LD snippet in Listing 1 below, describing a simple RO-Crate containing data entities described using contextual entities:

[29]<https://www.researchobject.org/ro-crate/1.2-DRAFT/profiles>

[30]<https://w3id.org/workflowhub/workflow-ro-crate/>

[31]<https://json-ld.org/>

[32]<https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html>

[33]The avid reader may spot that the RO-Crate Metadata file use the extension `.json` instead of `.jsonld`, this is to emphasize the developer expectations as a JSON format, while the file's JSON-LD nature is secondary. See https://github.com/ResearchObject/ro-crate/issues/82

```json
{ "@context": "https://w3id.org/ro/crate/1.1/context",
  "@graph": [
      { "@id": "ro-crate-metadata.json",
        "@type": "CreativeWork",
        "conformsTo": {"@id": "https://w3id.org/ro/crate/1.1"},
        "about": {"@id": "./"}
      },
      { "@id": "./",
        "@type": "Dataset",
        "name": "A simplified RO-Crate",
        "author": {"@id": "#alice"},
        "license": {"@id": "https://spdx.org/licenses/CC-BY-4.0"},
        "datePublished": "2021-11-02T16:04:43Z",
        "hasPart": [
          {"@id": "survey-responses-2019.csv"},
          {"@id": "https://example.com/pics/5707039334816454031_o.jpg"}
        ]
      },
      { "@id": "survey-responses-2019.csv",
        "@type": "File",
        "about": {"@id": "https://example.com/pics/5707039334816454031_o.jpg"},
        "author": {"@id": "#alice"}
      },
      { "@id": "https://example.com/pics/5707039334816454031_o.jpg",
        "@type": ["File", "ImageObject"],
        "contentLocation": {"@id": "http://sws.geonames.org/8152662/"},
        "author": {"@id": "https://orcid.org/0000-0002-1825-0097"}
      },
      { "@id": "#alice",
        "@type": "Person",
        "name": "Alice"
      },
      { "@id": "https://orcid.org/0000-0002-1825-0097",
        "@type": "Person",
        "name": "Josiah Carberry"
      },
      { "@id": "http://sws.geonames.org/8152662/",
        "@type": "Place",
        "name": "Catalina Park"
      },
      { "@id": "https://spdx.org/licenses/CC-BY-4.0",
        "@type": "CreativeWork",
        "name": "Creative Commons Attribution 4.0"
      }
  ]
}
```

Listing 1: Simplified[34] RO-Crate metadata file showing the flattened compacted JSON-LD `@graph` array containing the data entities and contextual entities, cross-referenced using `@id`. The `ro-crate-metadata.json` entity self-declares conformance with the RO-Crate specification using a versioned persistent identifier, further RO-Crate descriptions are on the root data entity `./` or any of the referenced data or contextual entities, as exemplified by the data entity `ImageObject` referencing contextual entities for `contentLocation` and `author` that differs from that of the overall RO-Crate. In this example `about` of the CSV data entity reference the `ImageObject`, which then take the roles of both a data entity and contextual entity. While `Person` entities ideally are identified with ORCID PIDs as for Josiah, in contrast `#alice` is here an RO-Crate local identifier, highlighting the pragmatic "just enough" Linked Data approach.

In this flattened profile of JSON-LD, each `{entity}` are directly under `@graph` and represents the RDF triples with a common *subject* (`@id`), mapped *properties* like `hasPart`, and *objects* — as either literal `"string"` values, referenced `{objects}` (which properties are listed in its own entity), or a JSON `[list]` of these. If processed as JSON-LD, this forms an RDF graph by matching the `@id` IRIs and applying the `@context` mapping to schema.org terms.

2.3.2. Flattened JSON-LD

When JSON-LD 1.0 [32] was proposed, one of the motivations was to seamlessly apply an RDF nature on top of regular JSON as frequently used by Web APIs. JSON objects in APIs are frequently nested with objects at multiple levels, and the perhaps most common form of JSON-LD is the compacted form[35] which follows this expectation (JSON-LD 1.1[36] further expands these capabilities, e.g. allowing nested `@context` definitions).

While this feature of JSON-LD can be seen as a way to "hide" its RDF nature, we found that the use of nested trees (e.g. a `Person` entity appearing as `author` of a `File` which nests under a `Dataset` with `hasPart`) counter-intuitively forces consumers to consider the JSON-LD as an RDF Graph, since an identified `Person` entity can appear at multiple and repeated points of the tree (e.g. author of multiple files), necessitating node merging or duplication, which can become complicated as this approach also invites the use of *blank nodes* (entities missing `@id`).

By comparison, a single flat `@graph` array approach, as required by RO-Crate, means that applications can choose to process and edit each entity as pure JSON by a simple lookup based on `@id`. At the same time, lifting all entities to the same level reflects the Research Object principles [11] in that describing the context and provenance is just as important as describing the data, and the requirement of `@id` of every entity forces RO-Crate generators to consciously consider existing IRIs and identifiers[37].

2.3.3. JSON-LD context

In JSON-LD, the `@context` is a reference to another JSON-LD document that provides mapping from JSON keys to Linked Data term IRIs, and can enable various JSON-LD directives to cater for customized JSON structures for translating to RDF.

[34]Recommended properties for types shown in Listing 1 also include `affiliation`, `citation`, `contactPoint`, `description`, `encodingFormat`, `funder`, `geo`, `identifier`, `keywords`, `publisher`; these properties and corresponding contextual entities are excluded here for brevity. See https://www.researchobject.org/2021-packaging-research-artefacts-with-ro-crate/listing1/

[35]<https://json-ld.org/spec/REC/json-ld/20140116/#compacted-document-form>

[36]<https://www.w3.org/TR/2020/REC-json-ld11-20200716/>

[37]<https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#describing-entities-in-json-ld>

RO-Crate reuses vocabulary terms and IRIs from Schema.org, but provides its own versioned JSON-LD context[38], which has a flat list with the mapping from JSON-LD keys to their URI equivalents (e.g. key `"author"` maps to the http://schema.org/author property).

The rationale behind this decision is to support JSON-based RO-Crate applications that are largely unaware of JSON-LD, that still may want to process the `@context` to find or add Linked Data definitions of otherwise unknown properties and types. Not reusing the official Schema.org context means RO-Crate is also able to map in additional vocabularies where needed, namely the *Portland Common Data Model* (PCDM) [36] for repositories and Bioschemas [37] for describing computational workflows. RO-Crate profiles may extend[39] the `@context` to re-use additional domain-specific ontologies.

Similarly, while the Schema.org context currently[40] have `"@type": "@id"` annotations for object properties, RO-Crate JSON-LD distinguishes explicitly between references to other entities (`{"@id": "#alice"}`) and string values (`"Alice"`) — meaning RO-Crate applications can find references for corresponding entities and IRIs without parsing the `@context` to understand a particular property. Notably this is exploited by the `ro-crate-html-js` [38] tool to provide reliable HTML rendering for otherwise unknown properties and types.

2.4. RO-Crate Community

The RO-Crate conceptual model, implementation and best practices are developed by a growing community of researchers, developers and publishers. RO-Crate's community is a key aspect of its effectiveness in making research artefacts FAIR. Fundamentally, the community provides the overall context of the implementation and model and ensures its interoperability.

The RO-Crate community consists of:

1. a diverse set of people representing a variety of stakeholders;
2. a set of collective norms;
3. an open platform that facilitates communication (GitHub, Google Docs, monthly teleconferences).

2.4.1. People

The initial concept of RO-Crate was formed at the first Workshop on Research Objects (RO2018[41]), held as part of the IEEE conference on eScience. This workshop followed up on considerations made at a Research Data Alliance (RDA) meeting on Research Data Packaging[42] that found similar goals across multiple data packaging efforts [13]: simplicity, structured metadata and the use of JSON-LD.

An important outcome of discussions that took place at RO2018 was the conclusion that the original Wf4Ever Research Object ontologies [12], in principle sufficient for packaging research artefacts with rich descriptions, were, in practice, considered inaccessible for regular programmers (e.g., Web developers) and in danger of being incomprehensible for domain scientists due to their reliance on Semantic Web technologies and other ontologies.

[38]<https://w3id.org/ro/crate/1.1/context>
[39]<https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#extending-ro-crate>
[40]<https://schema.org/version/13.0/schemaorg-current-http.jsonld>
[41]<https://www.researchobject.org/ro2018/>
[42]<https://rd-alliance.org/approaches-research-data-packaging-rda-11th-plenary-bof-meeting>

DataCrate [39] was presented at RO2018 as a promising lightweight alternative approach, and an agreement was made by a group of volunteers to attempt building what was initially called *"RO Lite"* as a combination of DataCrate's implementation and Research Object's principles.

This group, originally made up of library and Semantic Web experts, has subsequently grown to include domain scientists, developers, publishers and more. This perspective of multiple views led to the specification being used in a variety of domains, from bioinformatics and regulatory submissions to humanities and cultural heritage preservation.

The RO-Crate community is strongly engaged with the European-wide biology/bioinformatics collaborative e-Infrastructure, ELIXIR [40], along with European Open Science Cloud (EOSC[43]) projects including EOSC-Life[44], FAIRplus[45], CS3MESH4EOSC[46] and BY-COVID[47]. RO-Crate has also established collaborations with Bioschemas [37], GA4GH [41], OpenAIRE [42] and multiple H2020 projects.

A key set of stakeholders are developers: the RO-Crate community has made a point of attracting developers who can implement the specifications but, importantly, keeps "developer user experience" in mind. This means that the specifications are straightforward to implement and thus do not require expertise in technologies that are not widely deployed.

This notion of catering to "developer user experience" is an example of the set of norms that have developed and now define the community.

2.4.2. Norms

The RO-Crate community is driven by informal conventions and notions that are prevalent but not neccessarily written down. Here, we distil what we as authors believe are the critical set of norms that have facilitated the development of RO-Crate and contributed to the ability for RO-Crate research packages to be FAIR. This is not to say that there are no other norms within the community nor that everyone in the community holds these uniformly. Instead, what we emphasise is that these norms are helpful and also shaped by community practices.

1. Simplicity
2. Developer friendliness
3. Focus on examples and best practices rather than rigorous specification
4. Reuse "just enough" Web standards

A core norm of RO-Crate is that of **simplicity**, which sets the scene for how we guide developers to structure metadata with RO-Crate. We focus mainly on documenting simple approaches to the most common use cases, such as authors having an affiliation. This norm also influences our take on **developer friendliness**; for instance, we are using the Web-native JSON format, allowing only a few of JSON-LD's flexible Linked Data features. Moreover, the RO-Crate documentation is largely built up by **examples** showcasing **best practices**, rather than rigorous specifications. We build on existing **Web standards** that themselves are defined rigorously, which we utilise ***"just enough"*** in order to benefit from the advantages of Linked Data (e.g., extensions by namespaced vocabularies), without imposing too many developer choices or uncertainties (e.g., having to choose between the many RDF syntaxes).

[43]<https://eosc.eu/>
[44]<https://www.eosc-life.eu/>
[45]<https://fairplus-project.eu/>
[46]<https://cs3mesh4eosc.eu/>
[47]<https://by-covid.eu/>

While the above norms alone could easily lead to the creation of “yet another” JSON format, we keep the goal of **FAIR interoperability** of the captured metadata, and therefore follow closely FAIR best practices and current developments such as data citations, PIDs, open repositories and recommendations for sharing research outputs and software.

2.4.3. Open Platforms

The critical infrastructure that enables the community around RO-Crate is the use of open development platforms. This underpins the importance of open community access to supporting FAIR. Specifically, it is difficult to build and consume FAIR research artefacts without being able to access the specifications, understand how they are developed, know about any potential implementation issues, and discuss usage to evolve best practices.

The development of RO-Crate was driven by capturing documentation of real-life examples and best practices rather than creating a rigorous specification. At the same time, we agreed to be opinionated on the syntactic form to reduce the jungle of implementation choices; we wanted to keep the important aspects of Linked Data to adhere to the FAIR principles while retaining the option of combining and extending the structured metadata using the existing Semantic Web stack, not just build a standalone JSON format.

Further work during 2019 started adapting the DataCrate documentation through a more collaborative and exploratory *RO Lite* phase, initially using Google Docs for review and discussion, then moving to GitHub as a collaboration space for developing what is now the RO-Crate specification, maintained as Markdown[48] in GitHub Pages and published through Zenodo.

In addition to the typical Open Source-style development with GitHub issues and pull requests, the RO-Crate Community have, at time of writing, two regular monthly calls, a Slack channel and a mailing list for coordinating the project; also many of its participants collaborate on RO-Crate at multiple conferences and coding events such as the ELIXIR BioHackathon[49]. The community is jointly developing the RO-Crate specification and Open Source tools, as well as providing support and considering new use cases. The RO-Crate Community[50] is open for anyone to join, to equally participate under a code of conduct, and as of October 2021 has more than 50 members (see Appendix B).

3. RO-Crate Tooling

The work of the community has led to the development of a number of tools for creating and using RO-Crates. Table 1 shows the current set of implementations. Reviewing this list, one can see that tools support commonly used programming languages, including Python, JavaScript, and Ruby. Additionally, these tools can be integrated into commonly used research environments, in particular, the command line tool *ro-crate-html-js* [38] for creating a human-readable preview of an RO-Crate as a sidecar HTML file. Furthermore, there are tools that cater to end-users (*Describo* [22], *WorkflowHub* [48]), in order to simplify creating and managing RO-Crate. For example, Describo was developed to help researchers of the Australian Criminal Characters project[51] annotate historical prisoner records to gain greater insight into the history of Australia [61].

[48]<https://github.com/researchobject/ro-crate/>
[49]<https://biohackathon-europe.org/>
[50]<https://www.researchobject.org/ro-crate/community>
[51]<https://criminalcharacters.com/>

Tool name	Targets	Language / Platform	Status
Brief Description			
Describo [22]	Research Data Managers	NodeJS (Desktop)	RC
Interactive desktop application to create, update and export RO-Crates for different profiles			
Describo Online [43]	Platform developers	NodeJS (Web)	Alpha
Web-based application to create RO-Crates using cloud storage			
ro-crate-excel [44]	Data managers	JavaScript	
Command-line tool to create/edit RO-Crates with spreadsheets			
ro-crate-html-js [38]	Developers	JavaScript	Beta
HTML rendering of RO-Crate			
ro-crate-js [45]	Research Data Managers	JavaScript	Alpha
Library for creating/manipulating crates; basic validation code			
ro-crate-ruby [46]	Developers	Ruby	Beta
Ruby library for reading/writing RO-Crate, with workflow support			
ro-crate-py [47]	Developers	Python	Beta
Object-oriented Python library for reading/writing RO-Crate and use by Jupyter Notebook			
WorkflowHub [48]	Workflow users	Ruby	Beta
Workflow repository; imports and exports Workflow RO-Crate			
Life Monitor [49]	Workflow developers	Python	Alpha
Workflow testing and monitoring service; Workflow Testing profile of RO-Crate			
SCHeMa [50]	Workflow users	PHP	Alpha
Workflow execution using RO-Crate as exchange mechanism			
galaxy2cwl [51]	Workflow developers	Python	Alpha
Wraps Galaxy workflow as Workflow RO-Crate			
Modern PARADISEC [52]	Repository managers	Platform	Beta
Cultural Heritage portal based on OCFL and RO-Crate			
ONI express [53]	Repository managers	Platform	Beta
Platform for publishing data and documents stored in an OCFL repository via a Web interface			
ocfl-tools [54]	Developers	JavaScript (CLI)	Beta
Tools for managing RO-Crates in an OCFL repository			
RO Composer [55]	Repository developers	Java	Alpha
REST API for gradually building ROs for given profile			
RDA maDMP Mapper [56]	Data Management Plan users	Python	Beta
Mapping between machine-actionable data management plans (maDMP) and RO-Crate [57]			
Ro-Crate_2_ma-DMP [58]	Data Management Plan users	Python	Beta
Convert between machine-actionable data management plans (maDMP) and RO-Crate			
CheckMyCrate [59]	Developers	Python (CLI)	Alpha
Validation according to Workflow RO-Crate profile			
RO-Crates-and-Excel [60]	Data Managers	Java (CLI)	Alpha
Describe column/data details of spreadsheets as RO-Crate using DataCube vocabulary			

Table 1

Applications and libraries implementing RO-Crate, targeting different types of users across multiple programming languages. Status is indicative as assessed by this work (Alpha < Beta < Release Candidate (RC) < Release).

While the development of these tools is promising, our analysis of their maturity status shows that the majority of them are in the Beta stage. This is partly due to the fact that the RO-Crate specification itself only recently reached 1.0 status, in November 2019 [62]. Now that there is a fixed point of reference: With version 1.1 (October 2020) [63] RO-Crate has stabilised based on feedback from application development, and now we are seeing a further increase in the maturity of these tools, along with the creation of new ones.

Given the stage of the specification, these tools have been primarily targeting developers, essentially providing them with the core libraries for working with RO-Crate. Another target has been that of research data managers who need to manage and curate large amounts of data.

4. Profiles of RO-Crate in use

RO-Crate fundamentally forms part of an infrastructure to help build FAIR research artefacts. In other words, the key question is whether RO-Crate can be used to share and (re)use research artefacts. Here we look at three research domains where RO-Crate is being applied: Bioinformatics, Regulatory Science and Cultural Heritage. In addition, we note how RO-Crate may have an important role as part of machine-actionable data management plans and institutional repositories.

From these varied uses of RO-Crate we observe natural differences in their detail level and the type of entities described by the RO-Crate. For instance, on submission of an RO-Crate to a workflow repository, it is reasonable to expect the RO-Crate to contain at least one workflow, ideally with a declared licence and workflow language. Specific additional recommendations such as on identifiers is also needed to meet the emerging requirements of FAIR Digital Objects[52]. Work has now begun[53] to formalise these different *profiles* of RO-Crates, which may impose additional constraints based on the needs of a specific domain or use case.

4.1. Bioinformatics workflows

WorkflowHub.eu[54] is a European cross-domain registry of computational workflows, supported by European Open Science Cloud projects, e.g. EOSC-Life[55], and research infrastructures including the pan-European bioinformatics network ELIXIR[56] [40]. As part of promoting workflows as reusable tools, WorkflowHub includes documentation and high-level rendering of the workflow structure independent of its native workflow definition format. The rationale is that a domain scientist can browse all relevant workflows for their domain, before narrowing down their workflow engine requirements. As such, the WorkflowHub is intended largely as a registry of workflows already deposited in repositories specific to particular workflow languages and domains, such as UseGalaxy.eu [64] and Nextflow nf-core [65].

We here describe three different RO-Crate profiles developed for use with WorkflowHub.

[52]<https://fairdo.org/>
[53]<https://github.com/ResearchObject/ro-crate/issues/153>
[54]<https://workflowhub.eu/>
[55]<https://www.eosc-life.eu/>
[56]<https://elixir-europe.org/>

4.1.1. Profile for describing workflows

Being cross-domain, WorkflowHub has to cater for many different workflow systems. Many of these, for instance Nextflow [66] and Snakemake [67], by virtue of their script-like nature, reference multiple neighbouring files typically maintained in a GitHub repository. This calls for a data exchange method that allows keeping related files together. WorkflowHub has tackled this problem by adopting RO-Crate as the packaging mechanism [68], typing and annotating the constituent files of a workflow and — crucially — marking up the workflow language, as many workflow engines use common file extensions like `*.xml` and `*.json`. Workflows are further described with authors, license, diagram previews and a listing of their inputs and outputs. RO-Crates can thus be used for interoperable deposition of workflows to WorkflowHub, but are also used as an archive for downloading workflows, embedding metadata registered with the WorkflowHub entry and translated workflow files such as abstract Common Workflow Language (CWL) [69] definitions and diagrams [70].

RO-Crate acts therefore as an interoperability layer between registries, repositories and users in WorkflowHub. The iterative development between WorkflowHub developers and the RO-Crate community heavily informed the creation of the Bioschemas [37] profile for Computational Workflows[57], which again informed the RO-Crate 1.1 specification on workflows[58] and led to the RO-Crate Python library [47] and WorkflowHub's **Workflow RO-Crate profile,**[59] which, in a similar fashion to RO-Crate itself, recommends which workflow resources and descriptions are required. This co-development across project boundaries exemplifies the drive for simplicity and for establishing best practices.

4.1.2. Profile for recording workflow runs

RO-Crates in WorkflowHub have so far been focused on workflows that are ready to be run, and development of WorkflowHub is now creating a **Workflow Run RO-Crate profile** for the purposes of benchmarking, testing and executing workflows. As such, RO-Crate serves as a container of both a *workflow definition* that may be executed and of a particular *workflow execution with test results.*

This workflow run profile is a continuation of our previous work with capturing workflow provenance in a Research Object in CWLProv [35] and TavernaPROV [71]. In both cases, we used the PROV Ontology [59], including details of every task execution with all the intermediate data, which required significant workflow engine integration.[60]

Simplifying from the CWLProv approach, the planned Workflow Run RO-Crate profile will use a high level Schema.org provenance[61] for the input/output boundary of the overall workflow execution. This *Level 1 workflow provenance* [35] can be expressed generally across workflow languages with minimal workflow engine changes, with the option of more detailed provenance traces as separate PROV artefacts in the RO-Crate as data entities. In the current development of Specimen Data Refinery[62] [73] these RO-Crates will document the text recognition workflow runs of digitised biological specimens, exposed as FAIR Digital Objects [74].

[57]<https://bioschemas.org/profiles/ComputationalWorkflow/1.0-RELEASE/>

[58]<https://www.researchobject.org/ro-crate/1.1/workflows.html>

[59]<https://about.workflowhub.eu/Workflow-RO-Crate/>

[60]CWLProv and TavernaProv predate RO-Crate, but use RO-Bundle[72], a similar Research Object packaging method with JSON-LD metadata.

[61]<https://www.researchobject.org/ro-crate/1.1/provenance.html#software-used-to-create-files>

[62]<https://github.com/DiSSCo/SDR>

WorkflowHub has recently enabled minting of Digital Object Identifiers (DOIs), a PID commonly used for scholarly artefacts, for registered workflows, e.g. `10.48546/workflowhub.workflow.56.1` [75], lowering the barrier for citing workflows as computational methods along with their FAIR metadata – captured within an RO-Crate. While it is not an aim for WorkflowHub to be a repository of workflow runs and their data, RO-Crates of *exemplar workflow runs* serve as useful workflow documentation, as well as being an exchange mechanism that preserves FAIR metadata in a diverse workflow execution environment.

4.1.3. Profile for testing workflows

The value of computational workflows, however, is potentially undermined by the ”collapse” over time of the software and services they depend upon: for instance, software dependencies can change in a non-backwards-compatible manner, or active maintenance may cease; an external resource, such as a reference index or a database query service, could shift to a different URL or modify its access protocol; or the workflow itself may develop hard-to-find bugs as it is updated. This *workflow decay* can take a big toll on the workflow's reusability and on the reproducibility of any processes it evokes [76].

For this reason, WorkflowHub is complemented by a monitoring and testing service called LifeMonitor[49], also supported by EOSC-Life. LifeMonitor's main goal is to assist in the creation, periodic execution and monitoring of workflow tests, enabling the early detection of software collapse in order to minimise its detrimental effects. The communication of metadata related to workflow testing is achieved through the adoption of a **Workflow Testing RO-Crate profile**[63] stacked on top of the *Workflow RO-Crate* profile. This further specialisation of Workflow RO-Crate allows to specify additional testing-related entities (test suites, instances, services, etc.), leveraging RO-Crate's extension mechanism[64] through the addition of terms from custom namespaces.

In addition to showcasing RO-Crate's extensibility, the testing profile is an example of the format's flexibility and adaptability to the different needs of the research community. Though ultimately related to a computational workflow, in fact, most of the testing-specific entities are more about describing a protocol for interacting with a monitoring service than a set of research outputs and its associated metadata. Indeed, one of LifeMonitor's main functionalities is monitoring and reporting on test suites running on existing Continuous Integration (CI) services, which is described in terms of service URLs and job identifiers in the testing profile. In principle, in this context, data could disappear altogether, leading to an RO-Crate consisting entirely of contextual entities. Such an RO-Crate acts more as an exchange format for communication between services (WorkflowHub and LifeMonitor) than as an aggregator for research data and metadata, providing a good example of the format's high versatility.

4.2. Regulatory Sciences

BioCompute Objects[65] (BCO) [77] is a community-led effort to standardise submissions of computational workflows to biomedical regulators. For instance, a genomics sequencing pipeline, as part of a personalised cancer treatment study, can be submitted to the US Food and Drugs Administration (FDA) for approval. BCOs are formalised in the standard IEEE 2791-2020 [78] as a combination

[63]<https://crs4.github.io/life_monitor/workflow_testing_ro_crate>
[64]<https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#extending-ro-crate>
[65]<https://biocomputeobject.org/>

of JSON Schemas[66] that define the structure of JSON metadata files describing exemplar workflow runs in detail, covering aspects such as the usability and error domain of the workflow, its runtime requirements, the reference datasets used and representative output data produced.

BCOs provide a structured view over a particular workflow, informing regulators about its workings independently of the underlying workflow definition language. However, BCOs have only limited support for additional metadata.[67] For instance, while the BCO itself can indicate authors and contributors, and in particular regulators and their review decisions, it cannot describe the provenance of individual data files or workflow definitions.

As a custom JSON format, BCOs cannot be extended with Linked Data concepts, except by adding an additional top-level JSON object formalised in another JSON Schema. A BCO and workflow submitted by upload to a regulator will also frequently consist of multiple cross-related files. Crucially, there is no way to tell whether a given `*.json` file is a BCO file, except by reading its content and check for its `spec_version`.

We can then consider how a BCO and its referenced artefacts can be packaged and transferred following FAIR principles. **BCO RO-Crate**[68][79], part of the BioCompute Object user guides, defines a set of best practices for wrapping a BCO with a workflow, together with its exemplar outputs in an RO-Crate, which then provides typing and additional provenance metadata of the individual files, workflow definition, referenced data and the BCO metadata itself.

Here the BCO is responsible for describing the *purpose* of a workflow and its run at an abstraction level suitable for a domain scientist, while the more open-ended RO-Crate describes the surroundings of the workflow, classifying and relating its resources and providing provenance of their existence beyond the BCO. This emerging *separation of concerns* is shown in Figure 3, and highlights how RO-Crate is used side-by-side of existing standards and tooling, even where there are apparent partial overlaps.

A similar separation of concerns can be found if considering the RO-Crate as a set of files, where the *transport-level* metadata, such as checksum of files, are delegated to separate BagIt[69] manifests, a standard focusing on the preservation challenges of digital libraries [26]. As such, RO-Crate metadata files are not required to iterate all the files in their folder hierarchy, only those that benefit from being described.

Specifically, a BCO description alone is insufficient for reliable re-execution of a workflow, which would need a compatible workflow engine depending on the original workflow definition language, so IEEE 2791 recommends using Common Workflow Language (CWL) [69] for interoperable pipeline execution. CWL itself relies on tool packaging in software containers using Docker[70] or Conda[71]. Thus, we can consider BCO RO-Crate as a stack: transport-level manifests of files (BagIt), provenance, typing and context of those files (RO-Crate), workflow overview and purpose (BCO), interoperable workflow definition (CWL) and tool distribution (Docker).

[66]<https://opensource.ieee.org/2791-object/ieee-2791-schema/>
[67]IEEE 2791-2020 do permit user extensions in the *extension domain* by referencing additional JSON Schemas.
[68]<https://biocompute-objects.github.io/bco-ro-crate/>
[69]<https://www.researchobject.org/ro-crate/1.1/appendix/implementation-notes.html#adding-ro-crate-to-bagit>
[70]<https://www.docker.com/>
[71]<https://docs.conda.io/>

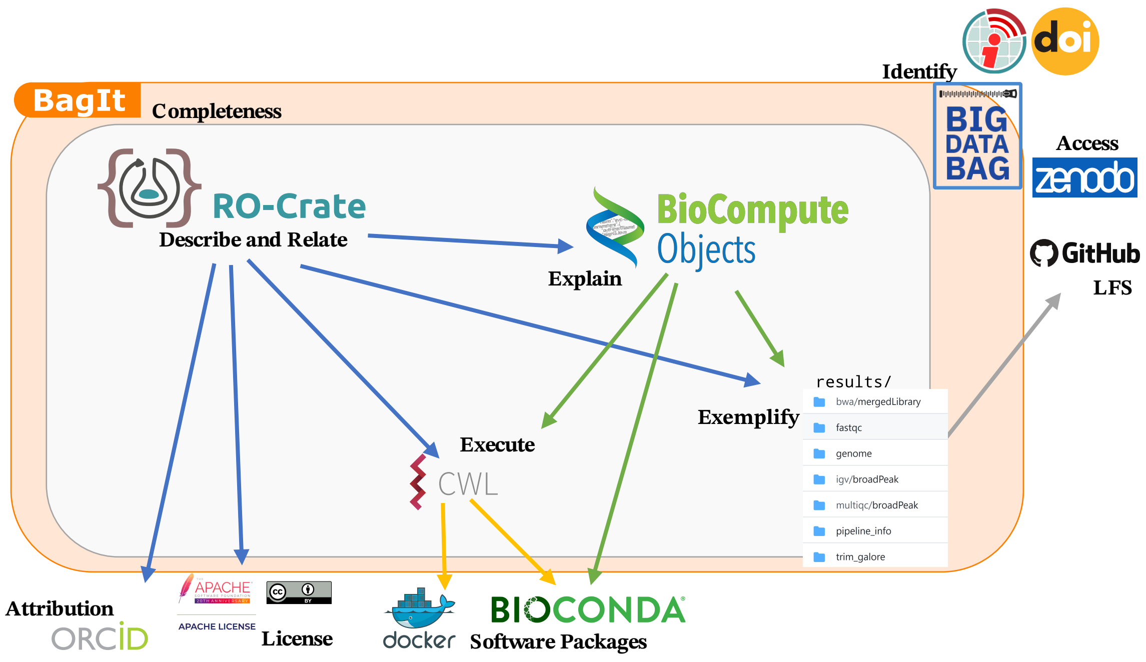

Fig. 3. **Separation of Concerns in BCO RO-Crate**. BioCompute Object (IEEE2791) is a JSON file that structurally explains the purpose and implementation of a computational workflow, for instance implemented in Common Workflow Language (CWL), that installs the workflow's software dependencies as Docker containers or BioConda packages. An example execution of the workflow shows the different kinds of result outputs, which may be external, using GitHub LFS [80] to support larger data. RO-Crate gathers all these local and external resources, relating them and giving individual descriptions, for instance permanent DOI identifiers for reused datasets accessed from Zenodo, but also adding external identifiers to attribute authors using ORCID or to identify which licences apply to individual resources. The RO-Crate and its local files are captured in a BagIt whose checksum ensures completeness, combined with Big Data Bag [81] features to "complete" the bag with large external files such as the workflow outputs.

4.3. Digital Humanities: Cultural Heritage

The Pacific And Regional Archive for Digital Sources in Endangered Cultures (PARADISEC[72]) [82] maintains a repository of more than 500,000 files documenting endangered languages across more than 16,000 items, collected and digitized over many years by researchers interviewing and recording native speakers across the region.

The Modern PARADISEC demonstrator[73] has been proposed[74] as an update to the 18 year old infrastructure, to also help long-term preservation of these artefacts in their digital form. The demonstrator uses RO-Crate to describe the overall structure and to capture the metadata of each item. The existing PARADISEC data collection has been ported and captured as RO-Crates. A Web portal then exposes the repository and its entries by indexing the RO-Crate metadata files, presenting a domain-specific view of the items — the RO-Crate is "hidden" and does not change the user interface.

[72]<https://www.paradisec.org.au/>
[73]<https://mod.paradisec.org.au/>
[74]<https://arkisto-platform.github.io/case-studies/paradisec/>

The PARADISEC use case takes advantage of several RO-Crate features and principles. Firstly, the transcribed metadata are now independent of the PARADISEC platform and can be archived, preserved and processed in its own right, using Schema.org as base vocabulary and extended with PARADISEC-specific terms.

In this approach, RO-Crate is the holder of itemised metadata, stored in regular files that are organised using Oxford Common File Layout[75] (OCFL) [27], which ensures file integrity and versioning on a regular shared file system. This lightweight infrastructure also gives flexibility for future developments and maintenance. For example a consumer can use Linked Data software such as a graph database and query the whole corpora using SPARQL triple patterns across multiple RO-Crates. For long term digital preservation, beyond the lifetime of PARADISEC portals, a "last resort" fallback is storing the generic RO-Crate HTML preview [38]. Such human-readable rendering of RO-Crates can be hosted as static files by any Web server, in line with the approach taken by the Endings Project.[76]

4.4. *Machine-actionable Data Management Plans*

Machine-actionable Data Management Plans (maDMPs) have been proposed as an improvement to automate FAIR data management tasks in research [83]; maDMPs use PIDs and controlled vocabularies to describe what happens to data over the research life cycle [84]. The Research Data Alliance's *DMP Common Standard* for maDMPs [85] is one such formalisation for expressing maDMPs, which can be expressed as Linked Data using the DMP Common Standard Ontology [86], a specialisation of the W3C Data Catalog Vocabulary (DCAT) [87]. RDA maDMPs are usually expressed using regular JSON, conforming to the DMP JSON Schema.

A mapping has been produced between Research Object Crates and Machine-actionable Data Management Plans [57], implemented by the RO-Crate RDA maDMP Mapper [56]. A similar mapping has been implemented by `RO-Crate_2_ma-DMP` [58]. In both cases, a maDMP can be converted to a RO-Crate, or vice versa. In [57] this functionality caters for two use cases:

1. Start a skeleton data management plan based on an existing RO-Crate dataset, e.g. from an RO-Crate from WorkflowHub.
2. Instantiate an RO-Crate based on a data management plan.

An important nuance here is that data management plans are (ideally) written in *advance* of data production, while RO-Crates are typically created to describe data *after* it has been generated. What is significant to note in this approach is the importance of **templating** in order to make both tasks automatable and achievable, and how RO-Crate can fit into earlier stages of the research life cycle.

4.5. *Institutional data repositories – Harvard Data Commons*

The concept of a **Data Commons** for research collaboration was originally defined as *"cyber-infrastructure that co-locates data, storage, and computing infrastructure with commonly used tools for analysing and sharing data to create an interoperable resource for the research community"* [88]. More recently, Data Commons has been established to mean integration of active data-intensive

[75]<https://ocfl.io/1.0/spec/>

[76]The Endings Project https://endings.uvic.ca/ is a five-year project funded by the Social Sciences and Humanities Research Council (SSHRC) that is creating tools, principles, policies and recommendations for digital scholarship practitioners to create accessible, stable, long-lasting resources in the humanities.

research with data management and archival best practices, along with a supporting computational infrastructure. Furthermore, the Commons features tools and services, such as computation clusters and storage for scalability, data repositories for disseminating and preserving regular, but also large or sensitive datasets, and other research assets. Multiple initiatives were undertaken to create Data Commons on national, research, and institutional levels. For example, the Australian Research Data Commons (ARDC)[77] [89] is a national initiative that enables local researchers and industries to access computing infrastructure, training, and curated datasets for data-intensive research. NCI's Genomic Data Commons[78] (GDC) [90] provides the cancer research community with access to a vast volume of genomic and clinical data. Initiatives such as Research Data Alliance (RDA) Global Open Research Commons[79] propose standards for the implementation of Data Commons to prevent them becoming ”data silos” and thus, enable interoperability from one Data Commons to another.

Harvard Data Commons [91] aims to address the challenges of data access and cross-disciplinary research within a research institution. It brings together multiple institutional schools, libraries, computing centres and the Harvard Dataverse[80] data repository. Dataverse[81] [92] is a free and open-source software platform to archive, share and cite research data. The Harvard Dataverse repository is the largest of 70 Dataverse installations worldwide, containing over 120K datasets with about 1.3M data files (as of 2021-11-16). Working toward the goal of facilitating collaboration and data discoverability and management within the university, Harvard Data Commons has the following primary objectives:

1. the integration of Harvard Research Computing with Harvard Dataverse by leveraging Globus endpoints [93]; this will allow an automatic transfer of large datasets to the repository. In some cases, only the metadata will be transferred while the data stays stored in remote storage;
2. support for advanced research workflows and providing packaging options for assets such as code and workflows in the Harvard Dataverse repository to enable reproducibility and reuse, and
3. interation of repositories supported by Harvard, which include DASH[82], the open access institutional repository, the Digital Repository Services (DRS) for preserving digital asset collections, and the Harvard Dataverse.

Particularly relevant to this article is the second objective of the Harvard Data Commons, which aims to support the deposit of research artefacts to Harvard Dataverse with sufficient information in the metadata to allow their future reuse (Figure 4). To support the incorporation of data, code, and other artefacts from various institutional infrastructures, Harvard Data Commons is currently working on RO-Crate adaptation. The RO-Crate metadata provides the necessary structure to make all research artefacts FAIR. The Dataverse software already has extensive support for metadata, including the Data Documentation Initiative (DDI), Dublin Core, DataCite, and Schema.org. Incorporating RO-Crate, which has the flexibility to describe a wide range of research resources, will facilitate their seamless transition from one infrastructure to the other within the Harvard Data Commons.

[77]<https://ardc.edu.au>
[78]<https://gdc.cancer.gov/>
[79]<https://www.rd-alliance.org/groups/global-open-research-commons-ig>
[80]<https://dataverse.harvard.edu/>
[81]<https://dataverse.org/>
[82]<https://dash.harvard.edu>

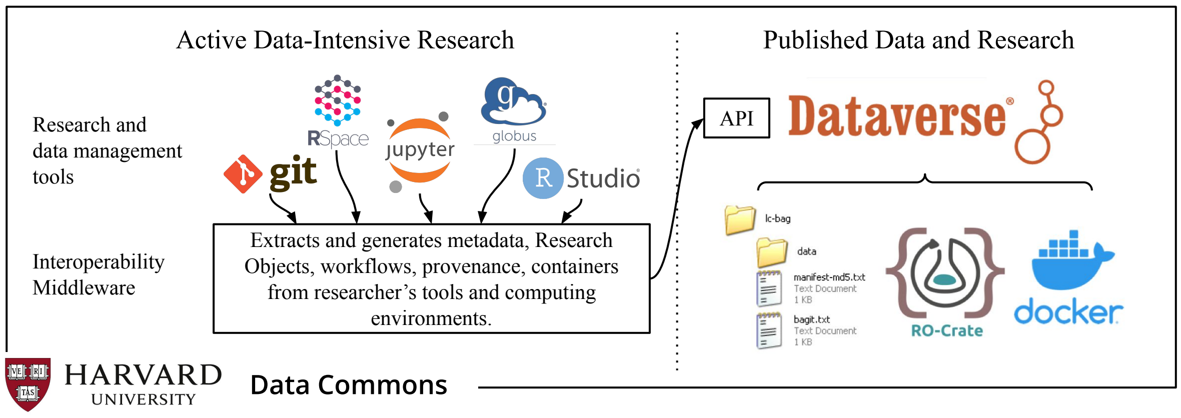
Active Data-Intensive Research
Published Data and Research
Research and data management tools
RSpace
jupyter
globus
git
RStudio
Interoperability Middleware
Extracts and generates metadata, Research Objects, workflows, provenance, containers from researcher's tools and computing environments.
API
Dataverse
lc-bag
data
manifest-md5.txt
bagit.txt
RO-Crate
docker
HARVARD UNIVERSITY
Data Commons

Fig. 4. **One aspect of Harvard Data Commons**. Automatic encapsulation and deposit of artefacts from data management tools used during active research at the Harvard Dataverse repository.

Even though the Harvard Data Commons is specific to Harvard University, the overall vision and the three objectives can be abstracted and applied to other universities or research organisations. The Commons will be designed and implemented using standards and commonly-used approaches to make it interoperable and reusable by others.

5. Related Work

With the increasing digitisation of research processes, there has been a significant call for the wider adoption of interoperable sharing of data and its associated metadata. We refer to [94] for a comprehensive overview and recommendations, in particular for data; notably that review highlights the wide variety of metadata and documentation that the literature prescribes for enabling data reuse. Likewise, we suggest [95] that covers the importance of metadata standards in reproducible computational research.

Here we focus on approaches for bundling research artefacts along with their metadata. This notion of publishing compound objects for scholarly communication has a long history behind it [96, 97], but recent approaches have followed three main strands: 1) publishing to centralised repositories; 2) packaging approaches similar to RO-Crate; and 3) bundling the computational workflow around a scientific experiment.

5.1. Bundling and Packaging Digital Research Artefacts

Early work making the case for publishing compound scholarly communication units [97] led to the development of the Object Re-Use and Exchange model[83] (OAI-ORE), providing a structured **resource map** of the digital artefacts that together support a scholarly output.

The challenge of describing computational workflows was one of the main motivations for the early proposal of *Research Objects* (RO) [11] as first-class citizens for sharing and publishing. The RO approach involves bundling datasets, workflows, scripts and results along with traditional dissemination materials like journal articles and presentations, forming a single package. Crucially, these

[83]<http://www.openarchives.org/ore/1.0/primer>

resources are not just gathered, but also individually typed, described and related to each other using semantic vocabularies. As pointed out in [11] an open-ended *Linked Data* approach is not sufficient for scholarly communication: a common data model is also needed in addition to common and best practices for managing and annotating lifecycle, ownership, versioning and attributions.

Considering the FAIR principles [5], we can say with hindsight that the initial RO approaches strongly targeted *Interoperability*, with a particular focus on the reproducibility of *in-silico experiments* involving computational workflows and the reuse of existing RDF vocabularies.

The first implementation of Research Objects for sharing workflows in myExperiment [98] was based on RDF ontologies [99], building on Dublin Core, FOAF, SIOC, Creative Commons and OAI-ORE to form myExperiment ontologies for describing social networking, attribution and credit, annotations, aggregation packs, experiments, view statistics, contributions, and workflow components [100].

This initially workflow-centric approach was further formalized as the Wf4Ever Research Object Model [12], which is a general-purpose research artefact description framework. This model is based on existing ontologies (FOAF, Dublin Core Terms, OAI-ORE and AO/OAC precursors to the W3C Web Annotation Model [101]) and adds specializations for workflow models and executions using W3C PROV-O [102]. The Research Object statements are saved in a *manifest* (the OAI-ORE *resource map*), with additional annotation resources containing user-provided details such as title and description.

We now claim that one barrier for wider adoption of the Wf4Eer Research Object model for general packaging digital research artefacts was exactly this re-use of multiple existing vocabularies (FAIR principle I2: *Metadata use vocabularies that follow FAIR principles*), which in itself is recogized as a challenge [103]. Adapters of the Wf4Ever RO model would have to navigate documentation of multiple overlapping ontologies, in addition to facing the usual Semantic Web development choices for RDF serialization formats, identifier minting and publishing resources on the Web.

Several developments for Research Objects improved on this situation, such as ROHub used by Earth Sciences [104], which provides a interactive user-interface for making Research Objects, along with Research Object Bundle [72] (RO Bundle), which is a ZIP-archive embedding data files and a JSON-LD serialization of the manifest with mappings for a limited set of terms. RO Bundle was also used for storing detailed workflow run provenance (TavernaPROV [71]).

RO-Bundle evolved to Research Object BagIt archives,[84] a variant of RO Bundle as a BagIt archive [26], used by Big Data Bags [81], CWLProv [35] and WholeTale [105, 106].

5.2. FAIR Digital Objects

FAIR Digital Objects (FDO) [74] have been proposed as a conceptual framework for making digital resources available in a Digital Objects (DO) architecture which encourages active use of the objects and their metadata. In particular, an FDO has five parts: (i) The FDO *content*, bit sequences stored in an accessible repository; (ii) a *Persistent Identifier* (PID) such as a DOI that identifies the FDO and can resolve these same parts; (iii) Associated rich *metadata*, as separate FDOs; (iv) Type definitions, also separate FDOs; (v) Associated *operations* for the given types. A Digital Object typed as a Collection aggregates other DOs by reference.

The Digital Object Interface Protocol [107] can be considered an "abstract protocol" of requirements, DOs could be implemented in multiple ways. One suggested implementation is the FAIR

[84]<https://w3id.org/ro/bagit>

Digital Object Framework[85], based on HTTP and the Linked Data Principles. While there is agreement on using PIDs based on DOIs, consensus on how to represent common metadata, core types and collections as FDOs has not yet been reached. We argue that RO-Crate can play an important role for FDOs:

1. By providing a predictable and extensible serialisation of structured metadata.
2. By formalising how to aggregate digital objects as collections (and adding their context).
3. By providing a natural Metadata FDO in the form of the RO-Crate Metadata File.
4. By being based on Linked Data and schema.org vocabulary, meaning that PIDs already exist for common types and properties.

At the same time, it is clear that the goal of FDO is broader than that of RO-Crate; namely, FDOs are active objects with distributed operations, and add further constraints such as PIDs for every element. These features improve FAIR features of digital objects and are also useful for RO-Crate, but they also severely restrict the infrastructure that needs to be implemented and maintained in order for FDOs to remain accessible. RO-Crate, on the other hand, is more flexible: it can minimally be used within any file system structure, or ideally exposed through a range of Web-based scenarios. A *FAIR profile of RO-Crate* (e.g. enforcing PID usage) will fit well within a FAIR Digital Object ecosystem.

5.3. Packaging Workflows

The use of computational workflows, typically combining a chain of tools in an analytical pipeline, has gained prominence in particular in the life sciences. Workflows might be used primarily to improve computational scalability, as well as to also assist in making computed data results FAIR [4], for instance by improving reproducibility [108], but also because programmatic data usage help propagate their metadata and provenance [109]. At the same time, workflows raise additional FAIR challenges, since they can be considered important research artefacts themselves. This viewpoint poses the problem of capturing and explaining the computational methods of a pipeline in sufficient machine-readable detail [3].

Even when researchers follow current best practices for workflow reproducibility [108, 110], the communication of computational outcomes through traditional academic publishing routes effectively adds barriers as authors are forced to rely on a textual manuscript representations. This hinder reproducibility and FAIR use of the knowledge previously captured in the workflow.

As a real-life example, let us look at a metagenomics article [111] that describes a computational pipeline. Here the authors have gone to extraordinary efforts to document the individual tools that have been reused, including their citations, versions, settings, parameters and combinations. The *Methods* section is two pages in tight double-columns with twenty four additional references, supported by the availability of data on an FTP server (60 GB) [112] and of open source code in GitHub Finn-Lab/MGS-gut[86] [113], including the pipeline as shell scripts and associated analysis scripts in R and Python.

This attention to reporting detail for computational workflows is unfortunately not yet the norm, and although bioinformatics journals have strong *data availability* requirements, they frequently do not require authors to include or cite *software, scripts and pipelines* used for analysing and producing results [114]. Indeed, in the absence of a specific requirement and an editorial policy to back it up

[85]<https://fairdigitalobjectframework.org/>
[86]<https://github.com/Finn-Lab/MGS-gut>

– such as eliminating the reference limit – authors are effectively discouraged from properly and comprehensively citing software [115].

However detailed this additional information might be, another researcher who wants to reuse a particular computational method may first want to assess if the described tool or workflow is Re-runnable (executable at all), Repeatable (same results for original inputs on same platform), Reproducible (same results for original inputs with different platform or newer tools) and ultimately Reusable (similar results for different input data), Repurposable (reusing parts of the method for making a new method) or Replicable (rewriting the workflow following the method description) [116, 117].

Following the textual description alone, researchers would be forced to jump straight to evaluate "Replicable" by rewriting the pipeline from scratch. This can be expensive and error-prone. They would firstly need to install all the software dependencies and download reference datasets. This can be a daunting task, which may have to be repeated multiple times as workflows typically are developed at small scale on desktop computers, scaled up to local clusters, and potentially put into production using cloud instances, each of which will have different requirements for software installations.

In recent years the situation has been greatly improved by software packaging and container technologies like Docker and Conda, these technologies have been increasingly adopted in life sciences [118] thanks to collaborative efforts such as BioConda [119] and BioContainers [120], and support by Linux distributions (e.g. Debian Med [121]). As of November 2021, more than 9,000 software packages are available in BioConda alone[87], and 10,000 containers in BioContainers[88].

Docker and Conda have been integrated into workflow systems such as Snakemake [67], Galaxy [122] and Nextflow [66], meaning a downloaded workflow definition can now be executed on a "blank" machine (except for the workflow engine) with the underlying analytical tools installed on demand. Even with using containers there is a reproducibility challenge, for instance Docker Hub's retention policy will expire container images after six months[89], or a lack of recording versions of transitive dependencies of Conda packages could cause incompatibilities if the packages are subsequently updated.

These container and package systems only capture small amounts of metadata[90]. In particular, they do not capture any of the semantic relationships between their content. Understanding these relationships is made harder by the opaque wrapping of arbitrary tools with unclear functionality, licenses and attributions.

From this we see that computational workflows are themselves complex digital objects that need to be recorded not just as files, but in the context of their execution environment, dependencies and analytical purpose in research – as well as other metadata (e.g. version, license, attribution and identifiers).

It is important to note that having all these computational details in order to represent them in an RO-Crate is an ideal scenario – in practice there will always be gaps of knowledge, and exposing all provenance details automatically would require improvements to the data sources, workflow,

[87]<https://anaconda.org/bioconda/>

[88]<https://biocontainers.pro/#/registry>

[89]<https://www.docker.com/blog/docker-hub-image-retention-policy-delayed-and-subscription-updates/>

[90]Docker and Conda can use *build recipes*, a set of commands that construct the container image through downloading and installing its requirements. However these recipes are effectively another piece of software code, which may itself decay and become difficult to rerun.

workflow engine and its dependencies. RO-Crate can be seen as a flexible annotation mechanism for augmenting automatic workflow provenance. Additional metadata can be added manually, e.g. for sensitive clinical data that cannot be publicly exposed[91], or to cite software that lack persistent identifiers. This inline *FAIRifying* allows researchers to achieve "just enough FAIR" to explain their computational experiments.

6. Conclusion

RO-Crate has been established as an approach to packaging digital research artefacts with structured metadata. This approach assists developers and researchers to produce and consume FAIR archives of their research.

RO-Crate is formed by a set of best practice recommendations, developed by an open and broad community. These guidelines show how to use ”just enough” Linked Data standards in a consistent way. The use of structured metadata with a rich base vocabulary can cover general-purpose contextual relations, with a Linked Data foundation that ensures extensibility to domain- and application-specific uses. We can therefore consider an RO-Crate not just as a structured data archive, but as a multimodal scholarly knowledge graph that can help ”FAIRify” and combine metadata of existing resources.

The adoption of simple Web technologies in the RO-Crate specification has helped a rapid development of a wide variety of supporting open source tools and libraries. RO-Crate fits into the larger landscape of open scholarly communication and FAIR Digital Object infrastructure, and can be integrated into data repository platforms. RO-Crate can be applied as a data/metadata exchange mechanism, assist in long-term archival preservation of metadata and data, or simply used at a small scale by individual researchers. Thanks to its strong community support, new and improved profiles and tools are being continuously added to the RO-Crate tooling landscape, making it easier for adopters to find examples and support for their own use case.

6.1. *Strictness vs flexibility*

There is always a tradeoff between flexibility and strictness [123] when deciding on semantics of metadata models. Strict requirements make it easier for users and code to consume and populate a model, by reducing choices and having mandated "slots" to fill in. But such rigidity can also restrict richness and applicability of the model, as it in turn enforce the initial assumptions about what can be described.

RO-Crate attempts to strike a balance between these tensions, and provides a common metadata framework that encourages extensions. However, just like the RO-Crate specification can be thought of as a *core profile* of schema.org in JSON-LD, we cannot stress the importance of also establishing domain-specific RO-Crate profiles and conventions, as explored in sections 2.2.6 and 4. Specialization comes hand-in-hand with the principle of *graceful degradation*; RO-Crate applications and users are free to choose the semantic detail level they participate at, as long as they follow the common syntactic requirements.

[91]FAIR principle A2: *Metadata are accessible, even when the data are no longer available.* [5]

7. Future Work

The direction of future RO-Crate work is determined by the community around it as a collaborative effort. We currently plan on further outreach, building training material (including a comprehensive entry-level tutorial) and maturing the reference implementation libraries. We will also collect and build examples of RO-Crate *consumption*, e.g. Jupyter Notebooks that query multiple crates using knowledge graphs. In addition, we are exploring ways to support some entity types requested by users, e.g. detailed workflow runs or container provenance, which do not have a good match in Schema.org. Such support could be added, for instance, by integrating other vocabularies or by having separated (but linked) metadata files.

Furthermore, we want to better understand how the community uses RO-Crate in practice and how it contrasts with other related efforts; this will help us to improve our specification and tools. By discovering commonalities in emerging usage (e.g. additional schema.org types), the community helps to reduce divergence that could otherwise occur with proliferation of further RO-Crate profiles. We plan to gather feedback via user studies, with the Linked Open Data community or as part of EOSC Bring-your-own-Data training events.

We operate in an open community where future and potential users of RO-Crate are actively welcomed to participate and contribute feedback and requirements. In addition we are targeting a wider audience through extensive outreach activities[92] and by initiating new connections. Recent contacts include American Geophysical Union (AGU) on Data Citation Reliquary [124], National Institute of Standards and Technology (NIST) on material science and InvenioRDM[93] used by the Zenodo data repository. New Horizon Europe projects adapting RO-Crate include BY-COVID[94], which aims to improve FAIR access to data on COVID-19 and other infectious diseases.

The main addition in the upcoming 1.2 release of the RO-Crate specifications will be the formalization of profiles[95] for different categories of crates. Additional entity types have been requested by users, e.g. workflow runs, business workflows, containers and software packages, tabular data structures; these are not always matched well with existing schema.org types but may benefit from other vocabularies or even separate metadata files, e.g. from Frictionless Data[96]. We will be further aligning with and collaborating with related research artefact description efforts like CodeMeta[97] for software metadata, Science-on-schema.org[98] [125] for datasets, FAIR Digital Objects[99] [@**(author?)** [74]] and activities in EOSC task forces[100] including the EOSC Interoperability Framework [16].

[92]<https://www.researchobject.org/ro-crate/outreach.html>
[93]<https://inveniosoftware.org/products/rdm/>
[94]<https://by-covid.org/>
[95]<https://www.researchobject.org/ro-crate/1.2-DRAFT/profiles>
[96]<https://frictionlessdata.io/>
[97]<https://codemeta.github.io/>
[98]<https://science-on-schema.org/>
[99]<https://fairdo.org/>
[100]<https://www.eosc.eu/task-force-faq>

8. Acknowledgements

This work has received funding from the European Commission's Horizon 2020 research and innovation programme for projects BioExcel-2[101] (H2020-INFRAEDI-2018-1 823830), IBISBA 1.0[102] (H2020-INFRAIA-2017-1-two-stage 730976), PREP-IBISBA[103] (H2020-INFRADEV-2019-2 871118), EOSC-Life[104] (H2020-INFRAEOSC-2018-2 824087), SyntheSys+[105] (H2020-INFRAIA-2018-1 823827). From the Horizon Europe Framework Programme this work has received funding for BY-COVID[106] (HORIZON-INFRA-2021-EMERGENCY-01 101046203).

Björn Grüning is supported by DataPLANT (NFDI 7/1 – 42077441[107]), part of the German National Research Data Infrastructure (NFDI), funded by the Deutsche Forschungsgemeinschaft (DFG).

Ana Trisovic is funded by the Alfred P. Sloan Foundation (grant number P-2020-13988)[108]. Harvard Data Commons is supported by an award from Harvard University Information Technology (HUIT).

8.1. Contributions

Author contributions to this article and the RO-Crate projet according to the Contributor Roles Taxonomy CASRAI CrEDiT[109] [126]:

Stian Soiland-Reyes Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Software, Visualization, Writing – original draft, Writing – review & editing
Peter Sefton Conceptualization, Investigation, Methodology, Project administration, Resources, Software, Writing – review & editing
Mercè Crosas Writing – review & editing
Leyla Jael Castro Methodology, Writing – review & editing
Frederik Coppens Writing – review & editing
José M. Fernández Methodology, Software, Writing – review & editing
Daniel Garijo Methodology, Writing – review & editing
Björn Grüning Writing – review & editing
Marco La Rosa Software, Methodology, Writing – review & editing
Simone Leo Software, Methodology, Writing – review & editing
Eoghan Ó Carragáin Investigation, Methodology, Project administration, Writing – review & editing
Marc Portier Methodology, Writing – review & editing
Ana Trisovic Software, Writing – review & editing
RO-Crate Community Investigation, Software, Validation, Writing – review & editing
Paul Groth Methodology, Supervision, Writing – original draft, Writing – review & editing
Carole Goble Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Visualization, Writing – review & editing

We would also like to acknowledge contributions from:

[101]<https://cordis.europa.eu/project/id/823830>
[102]<https://cordis.europa.eu/project/id/730976>
[103]<https://cordis.europa.eu/project/id/871118>
[104]<https://cordis.europa.eu/project/id/824087>
[105]<https://cordis.europa.eu/project/id/823827>
[106]<https://cordis.europa.eu/project/id/101046203>
[107]<https://gepris.dfg.de/gepris/projekt/442077441>
[108]<https://sloan.org/grant-detail/9555>
[109]<https://casrai.org/credit/>

Finn Bacall Software, Methodology
Herbert Van de Sompel Writing – review & editing
Ignacio Eguinoa Software, Methodology
Nick Juty Writing – review & editing
Oscar Corcho Writing – review & editing
Stuart Owen Writing – review & editing
Laura Rodríguez-Navas Software, Visualization, Writing – review & editing
Alan R. Williams Writing – review & editing

References

[1] P. Sefton, FAIR Data Management; It's a lifestyle not a lifecycle - ptsefton.com, 2021. http://ptsefton.com/2021/04/07/rdmpic/.

[2] V. Stodden, M. McNutt, D.H. Bailey, E. Deelman, Y. Gil, B. Hanson, M.A. Heroux, J.P.A. Ioannidis and M. Taufer, Enhancing reproducibility for computational methods, *Science* **354**(6317) (2016), 1240–1241. doi:10.1126/science.aah6168.

[3] A.-L. Lamprecht, L. Garcia, M. Kuzak, C. Martinez, R. Arcila, E. Martin Del Pico, V. Dominguez Del Angel, S. van de Sandt, J. Ison, P.A. Martinez, P. McQuilton, A. Valencia, J. Harrow, F. Psomopoulos, J.L. Gelpi, N. Chue Hong, C. Goble and S. Capella-Gutierrez, Towards FAIR principles for research software, *Déviance et société* (2019), 1–23. doi:10.3233/DS-190026.

[4] C. Goble, S. Cohen-Boulakia, S. Soiland-Reyes, D. Garijo, Y. Gil, M.R. Crusoe, K. Peters and D. Schober, FAIR Computational Workflows, *Data Intelligence* **2**(1–2) (2019), 108–121. doi:10.1162/dint_a_00033.

[5] M.D. Wilkinson, M. Dumontier, I.J.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.B. da Silva Santos, P.E. Bourne, J. Bouwman, A.J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C.T. Evelo, R. Finkers, A. Gonzalez-Beltran, A.J.G. Gray, P. Groth, C. Goble, J.S. Grethe, J. Heringa, P.A.C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S.J. Lusher, M.E. Martone, A. Mons, A.L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M.A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons, The FAIR Guiding Principles for scientific data management and stewardship., *Scientific data* **3** (2016), 160018. doi:10.1038/sdata.2016.18.

[6] B. Mons, *Data Stewardship for Open Science*, 1st edn, Taylor & Francis, p. 240. ISBN 9781315351148.

[7] M. Dillen, Q. Groom, D. Agosti and L. Nielsen, Zenodo, an Archive and Publishing Repository: A tale of two herbarium specimen pilot projects, *Biodiversity Information Science and Standards* **3** (2019). doi:10.3897/biss.3.37080.

[8] H. Berman, K. Henrick, H. Nakamura and J.L. Markley, The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data., *Nucleic Acids Research* **35**(Database issue) (2007), D301–3. doi:10.1093/nar/gkl971.

[9] G.K. Sandve, A. Nekrutenko, J. Taylor and E. Hovig, Ten simple rules for reproducible computational research., *PLoS Computational Biology* **9**(10) (2013), e1003285. doi:10.1371/journal.pcbi.1003285.

[10] L. Koesten, K. Gregory, P. Groth and E. Simperl, Talking datasets – Understanding data sensemaking behaviours, *International journal of human-computer studies* **146** (2021), 102562. doi:10.1016/j.ijhcs.2020.102562.

[11] S. Bechhofer, I. Buchan, D. De Roure, P. Missier, J. Ainsworth, J. Bhagat, P. Couch, D. Cruickshank, M. Delderfield, I. Dunlop, M. Gamble, D. Michaelides, S. Owen, D. Newman, S. Sufi and C. Goble, Why linked data is not enough for scientists, *Future Generation Computer Systems* **29**(2) (2013), 599–611. doi:10.1016/j.future.2011.08.004.

[12] K. Belhajjame, J. Zhao, D. Garijo, M. Gamble, K. Hettne, R. Palma, E. Mina, O. Corcho, J.M. Gómez-Pérez, S. Bechhofer, G. Klyne and C. Goble, Using a suite of ontologies for preserving workflow-centric research objects, *Web Semantics: Science, Services and Agents on the World Wide Web* **32**(0) (2015), 16–42. doi:10.1016/j.websem.2015.01.003.

[13] E.Ó. Carragáin, C. Goble, P. Sefton and S. Soiland-Reyes, A lightweight approach to research object data packaging, *Zenodo* (2019). doi:10.5281/zenodo.3250687.

[14] R.C. Amorim, J.A. Castro, J. Rocha da Silva and C. Ribeiro, A comparison of research data management platforms: architecture, flexible metadata and interoperability, *Universal Access in the Information Society* (2016), 1–12. doi:10.1007/s10209-016-0475-y.

[15] S. Farnel and A. Shiri, Metadata for Research Data: Current Practices and Trends, in: *2014 Proceedings of the International Conference on Dublin Core and Metadata Applications*, W. Moen and A. Rushing, eds, Dublin Core Metadata Initiative, 2014. ISSN 1939-1366. https://dcpapers.dublincore.org/pubs/article/view/3714.

[16] K. Kurowski, O. Corcho, C. Choirat, M. Eriksson, F. Coppens, M. van de Sanden and M. Ojsteršek, EOSC Interoperability Framework, Technical Report, 2021. doi:10.2777/620649.
[17] L.M. Chan, *Library of Congress Subject Headings: Principles and Application*, 3rd edn, Libraries Unlimited, Englewood, Colo, 1995, p. 556. ISBN 9781563081910. https://eric.ed.gov/?id=ED387146.
[18] M. Žumer, *National bibliographies in the digital age: guidance and new directions*, IFLA series on bibliographic control, Walter de Gruyter – K. G. Saur, 2009, IFLA Working Group on Guidelines for National Bibliographies. ISSN 1868-8438. ISBN 9783598441844. doi:10.1515/9783598441844.
[19] C. Neylon, As a researcher...I'm a bit bloody fed up with Data Management, 2017. https://cameronneylon.net/blog/as-a-researcher-im-a-bit-bloody-fed-up-with-data-management/.
[20] C.J. Volk, Y. Lucero and K. Barnas, Why is data sharing in collaborative natural resource efforts so hard and what can we do to improve it?, *Environmental Management* **53**(5) (2014), 883–893. doi:10.1007/s00267-014-0258-2.
[21] L.M. Schriml, M. Chuvochina, N. Davies, E.A. Eloe-Fadrosh, R.D. Finn, P. Hugenholtz, C.I. Hunter, B.L. Hurwitz, N.C. Kyrpides, F. Meyer, I.K. Mizrachi, S.-A. Sansone, G. Sutton, S. Tighe and R. Walls, COVID-19 pandemic reveals the peril of ignoring metadata standards., *Scientific data* **7**(1) (2020), 188. doi:10.1038/s41597-020-0524-5.
[22] M. La Rosa and P. Sefton, Arkisto Platform: Describo. https://arkisto-platform.github.io/describo/.
[23] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing and Jupyter Development Team, Jupyter Notebooks - a publishing format for reproducible computational workflows, in: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, IOS Press, 2016, pp. 87–90, Proceedings of the 20th International Conference on Electronic Publishing. ISBN 978-1-61499-649-1. doi:10.3233/978-1-61499-649-1-87.
[24] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, Vol. 1, 2011, pp. 1–136. ISSN 2160-4711. ISBN 9781608454310. doi:10.2200/S00334ED1V01Y201102WBE001.
[25] J.A. McMurry, N. Juty, N. Blomberg, T. Burdett, T. Conlin, N. Conte, M. Courtot, J. Deck, M. Dumontier, D.K. Fellows, A. Gonzalez-Beltran, P. Gormanns, J. Grethe, J. Hastings, J.-K. Hériché, H. Hermjakob, J.C. Ison, R.C. Jimenez, S. Jupp, J. Kunze, C. Laibe, N. Le Novère, J. Malone, M.J. Martin, J.R. McEntyre, C. Morris, J. Muilu, W. Müller, P. Rocca-Serra, S.-A. Sansone, M. Sariyar, J.L. Snoep, S. Soiland-Reyes, N.J. Stanford, N. Swainston, N. Washington, A.R. Williams, S.M. Wimalaratne, L.M. Winfree, K. Wolstencroft, C. Goble, C.J. Mungall, M.A. Haendel and H. Parkinson, Identifiers for the 21st century: How to design, provision, and reuse persistent identifiers to maximize utility and impact of life science data., *PLoS Biology* **15**(6) (2017), e2001414. doi:10.1371/journal.pbio.2001414.
[26] J. Kunze, J. Littman, E. Madden, J. Scancella and C. Adams, The BagIt File Packaging Format (V1.0), Technical Report, 2018. doi:10.17487/RFC8493. https://www.rfc-editor.org/info/rfc8493.
[27] OCFL, Oxford Common File Layout Specification, Recommendation, 2020. https://ocfl.io/1.0/spec/.
[28] R. Lammey, Solutions for identification problems: a look at the Research Organization Registry, *Science Editing* **7**(1) (2020), 65–69. doi:10.6087/kcse.192.
[29] C. Bizer, T. Heath and T. Berners-Lee, Linked data: the story so far, in: *Semantic services, interoperability and web applications: emerging concepts*, A. Sheth, ed., IGI Global, 2011, pp. 205–227. ISBN 9781609605933. doi:10.4018/978-1-60960-593-3.ch008.
[30] M. Duerst and M. Suignard, Internationalized resource identifiers (IRIs), Technical Report, 2005. doi:10.17487/rfc3987. https://www.rfc-editor.org/info/rfc3987.
[31] W3C Technical Architecture Group, Dereferencing HTTP URIs, Draft Tag Finding, 2007. https://www.w3.org/2001/tag/doc/httpRange-14/2007-08-31/HttpRange-14.html.
[32] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler and N. Lindström, JSON-LD 1.0, W3C Recommendation, 2014. https://www.w3.org/TR/2014/REC-json-ld-20140116/.
[33] P. Sefton, E.Ó. Carragáin, S. Soiland-Reyes, O. Corcho, D. Garijo, R. Palma, F. Coppens, C. Goble, J.M. Fernández, K. Chard, J.M. Gomez-Perez, M.R. Crusoe, I. Eguinoa, N. Juty, K. Holmes, J.A. Clark, S. Capella-Gutierrez, A.J.G. Gray, S. Owen, A.R. Williams, G. Tartari, F. Bacall, T. Thelen, H. Ménager, L.R.-N. Navas, P. Walk, B. Whitehead, M. Wilkinson, P. Groth, E. Bremer, L.G. Castro, K. Sebby, A. Kanitz, A. Trisovic, G. Kennedy, M. Graves, J. Koehorst, S. Leo and M. Portier, RO-Crate Metadata Specification 1.1.1, Technical Report, 2021. doi:10.5281/zenodo.4541002. https://w3id.org/ro/crate/1.1.
[34] R.V. Guha, D. Brickley and S. Macbeth, Schema.org: Evolution of Structured Data on the Web: Big data makes common schemas even more necessary, *Queue* **13**(9) (2015), 10–37. doi:10.1145/2857274.2857276.
[35] F.Z. Khan, S. Soiland-Reyes, R.O. Sinnott, A. Lonie, C. Goble and M.R. Crusoe, Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv, *GigaScience* **8**(11) (2019). doi:10.1093/gigascience/giz095.
[36] S. Cossu, E. Cowles, K. Estlund, C. Harlow, T. Johnson, M. Matienzo, D. Lamb, L. Rayle, R. Sanderson, J. Stroop and A. Woods, Portland Common Data Model, 2018. https://github.com/duraspace/pcdm/wiki.
[37] A. Gray, C. Goble, R. Jimenez and Bioschemas Community, Bioschemas: From Potato Salad to Protein Annotation, Vienna, Austria, 2017. https://iswc2017.semanticweb.org/paper-579/.

[38] npm: ro-crate-html-js. https://www.npmjs.com/package/ro-crate-html-js.
[39] P. Sefton, Datacrate Submisssion To The Workshop On Research Objects, *Zenodo* (2018). doi:10.5281/zenodo.1445817.
[40] L.C. Crosswell and J.M. Thornton, ELIXIR: a distributed infrastructure for European biological data, *Trends in Biotechnology* **30**(5) (2012), 241–242. doi:10.1016/j.tibtech.2012.02.002.
[41] H.L. Rehm, A.J.H. Page, L. Smith, J.B. Adams, G. Alterovitz, L.J. Babb, M.P. Barkley, M. Baudis, M.J.S. Beauvais, T. Beck, J.S. Beckmann, S. Beltran, D. Bernick, A. Bernier, J.K. Bonfield, T.F. Boughtwood, G. Bourque, S.R. Bowers, A.J. Brookes, M. Brudno, M.H. Brush, D. Bujold, T. Burdett, O.J. Buske, M.N. Cabili, D.L. Cameron, R.J. Carroll, E. Casas-Silva, D. Chakravarty, B.P. Chaudhari, S.H. Chen, J.M. Cherry, J. Chung, M. Cline, H.L. Clissold, R.M. Cook-Deegan, M. Courtot, F. Cunningham, M. Cupak, R.M. Davies, D. Denisko, M.J. Doerr, L.I. Dolman, E.S. Dove, L.J. Dursi, S.O.M. Dyke, J.A. Eddy, K. Eilbeck, K.P. Ellrott, S. Fairley, K.A. Fakhro, H.V. Firth, M.S. Fitzsimons, M. Fiume, P. Flicek, I.M. Fore, M.A. Freeberg, R.R. Freimuth, L.A. Fromont, J. Fuerth, C.L. Gaff, W. Gan, E.M. Ghanaim, D. Glazer, R.C. Green, M. Griffith, O.L. Griffith, R.L. Grossman, T. Groza, J.M. Guidry Auvil, R. Guigó, D. Gupta, M.A. Haendel, A. Hamosh, D.P. Hansen, R.K. Hart, D.M. Hartley, D. Haussler, R.M. Hendricks-Sturrup, C.W.L. Ho, A.E. Hobb, M.M. Hoffman, O.M. Hofmann, P. Holub, J.S. Hsu, J.-P. Hubaux, S.E. Hunt, A. Husami, J.O. Jacobsen, S.S. Jamuar, E.L. Janes, F. Jeanson, A. Jené, A.L. Johns, Y. Joly, S.J.M. Jones, A. Kanitz, K. Kato, T.M. Keane, K. Kekesi-Lafrance, J. Kelleher, G. Kerry, S.-S. Khor, B.M. Knoppers, M.A. Konopko, K. Kosaki, M. Kuba, J. Lawson, R. Leinonen, S. Li, M.F. Lin, M. Linden, X. Liu, I.U. Liyanage, J. Lopez, A.M. Lucassen, M. Lukowski, A.L. Mann, J. Marshall, M. Mattioni, A. Metke-Jimenez, A. Middleton, R.J. Milne, F. Molnár-Gábor, N. Mulder, M.C. Munoz-Torres, R. Nag, H. Nakagawa, J. Nasir, A. Navarro, T.H. Nelson, A. Niewielska, A. Nisselle, J. Niu, T.H. Nyrönen, B.D. O, S. Oesterle, S. Ogishima, V. Ota Wang, L.A.D. Paglione, E. Palumbo, H.E. Parkinson, A.A. Philippakis, A.D. Pizarro, A. Prlic, J. Rambla, A. Rendon, R.A. Rider, P.N. Robinson, K.W. Rodarmer, L.L. Rodriguez, A.F. Rubin, M. Rueda, G.A. Rushton, R.S. Ryan, G.I. Saunders, H. Schuilenburg, T. Schwede, S. Scollen, A. Senf, N.C. Sheffield, N. Skantharajah, A.V. Smith, H.J. Sofia, D. Spalding, A.B. Spurdle, Z. Stark, L.D. Stein, M. Suematsu, P. Tan, J.A. Tedds, A.A. Thomson, A. Thorogood, T.L. Tickle, K. Tokunaga, J. Törnroos, D. Torrents, S. Upchurch, A. Valencia, R.V. Guimera, J. Vamathevan, S. Varma, D.F. Vears, C. Viner, C. Voisin, A.H. Wagner, S.E. Wallace, B.P. Walsh, M.S. Williams, E.C. Winkler, B.J. Wold, G.M. Wood, J.P. Woolley, C. Yamasaki, A.D. Yates, C.K. Yung, L.J. Zass, K. Zaytseva, J. Zhang, P. Goodhand, K. North and E. Birney, GA4GH: International policies and standards for data sharing across genomic research and healthcare, *Cell Genomics* **1**(2) (2021), 100029. doi:10.1016/j.xgen.2021.100029.
[42] N. Rettberg and B. Schmidt, OpenAIRE, *College & Research Libraries News* **76**(6) (2015), 306–310. http://resolver.sub.uni-goettingen.de/purl?gs-1/11942.
[43] M. La Rosa, Arkisto Platform: Describo Online. https://arkisto-platform.github.io/describo-online/.
[44] M. Lynch and P. Sefton, npm: ro-crate-excel. https://www.npmjs.com/package/ro-crate-excel.
[45] GitHub - UTS-eResearch/ro-crate-js: Research Object Crate (RO-Crate) utilities. https://github.com/UTS-eResearch/ro-crate-js.
[46] F. Bacall and M. Whitwell, GitHub - ResearchObject/ro-crate-ruby: A Ruby gem for creating, manipulating and reading RO-Crates. https://github.com/ResearchObject/ro-crate-ruby.
[47] B. Droesbeke, I. Eguinoa, A. Gaignard, L. Simone, L. Pireddu, L. Rodríguez-Navas and S. Soiland-Reyes, GitHub - ResearchObject/ro-crate-py: Python library for RO-Crate. doi:10.5281/zenodo.3956493. https://github.com/researchobject/ro-crate-py.
[48] WorkflowHub project |Project pages for developing and running the WorkflowHub, a registry of scientific workflows. https://about.workflowhub.eu/.
[49] CRS4, LifeMonitor, a testing and monitoring service for scientific workflows. https://about.lifemonitor.eu/.
[50] T. Vergoulis, K. Zagganas, L. Kavouras, M. Reczko, S. Sartzetakis and T. Dalamagas, SCHeMa: Scheduling Scientific Containers on a Cluster of Heterogeneous Machines, *arXiv* (2021), 2103.13138–. https://arxiv.org/abs/2103.13138v1.
[51] GitHub - workflowhub-eu/galaxy2cwl: Standalone version tool to get cwl descriptions (initially an abstract cwl interface) of galaxy workflows and Galaxy workflows executions. https://github.com/workflowhub-eu/galaxy2cwl.
[52] GitHub - CoEDL/modpdsc. https://github.com/CoEDL/modpdsc/.
[53] Tools: Data Portal & Discovery. https://arkisto-platform.github.io/tools/portal/.
[54] GitHub - CoEDL/ocfl-tools: Tools to process and manipulate an OCFL tree. https://github.com/CoEDL/ocfl-tools.
[55] F. Bacall, S. Soiland-Reyes and M. Soares e Silva, eScienceLab: RO-Composer. https://esciencelab.org.uk/projects/ro-composer/.
[56] G. Arfaoui and M. Jaoua, RO-Crate RDA maDMP Mapper, *Zenodo* (2020). doi:10.5281/zenodo.3922136. https://github.com/GhaithArf/ro-crate-rda-madmp-mapper.
[57] T. Miksa, M. Jaoua and G. Arfaoui, Research Object Crates and Machine-actionable Data Management Plans, in: *1st Workshop on Research Data Management for Linked Open Science*, 2020. doi:10.4126/frl01-006423291.

[58] G. Brenner, BrennerG/Ro-Crate_2_ma-DMP: v1.0.0, 2020. doi:10.5281/zenodo.3903463. https://github.com/BrennerG/Ro-Crate_2_ma-DMP.

[59] K. Belchev, KockataEPich/CheckMyCrate: A command line application for validating a RO-Crate object against a JSON profile., GitHub, 2021. https://github.com/KockataEPich/CheckMyCrate.

[60] F. Zoubek and M. Winkler, RO Crates and Excel, Zenodo, 2021. doi:10.5281/zenodo.5068950. https://github.com/e11938258/RO-Crates-and-Excel.

[61] A. Piper, Digital crowdsourcing and public understandings of the past: citizen historians meet Criminal Characters, *History Australia* **17**(3) (2020), 525–541. doi:10.1080/14490854.2020.1796500.

[62] P. Sefton, E.Ó. Carragáin, S. Soiland-Reyes, O. Corcho, D. Garijo, R. Palma, F. Coppens, C. Goble, J.M. Fernández, K. Chard, J.M. Gomez-Perez, M.R. Crusoe, I. Eguinoa, N. Juty, K. Holmes, J.A. Clark, S. Capella-Gutierrez, A.J.G. Gray, S. Owen, A.R. Williams, G. Tartari, F. Bacall and T. Thelen, RO-Crate Metadata Specification 1.0, Technical Report, 2019. doi:10.5281/zenodo.3541888. https://w3id.org/ro/crate/1.0.

[63] P. Sefton, E.Ó. Carragáin, S. Soiland-Reyes, O. Corcho, D. Garijo, R. Palma, F. Coppens, C. Goble, J.M. Fernández, K. Chard, J.M. Gomez-Perez, M.R. Crusoe, I. Eguinoa, N. Juty, K. Holmes, J.A. Clark, S. Capella-Gutierrez, A.J.G. Gray, S. Owen, A.R. Williams, G. Tartari, F. Bacall, T. Thelen, H. Ménager, L. Rodríguez-Navas, P. Walk, B. Whitehead, M. Wilkinson, P. Groth, E. Bremer, L.G. Castro, K. Sebby, A. Kanitz, A. Trisovic, G. Kennedy, M. Graves, J. Koehorst and S. Leo, RO-Crate Metadata Specification 1.1, Technical Report, 2020. doi:10.5281/zenodo.4031327. https://w3id.org/ro/crate/1.1.

[64] D. Baker, M. van den Beek, D. Blankenberg, D. Bouvier, J. Chilton, N. Coraor, F. Coppens, I. Eguinoa, S. Gladman, B. Grüning, N. Keener, D. Larivière, A. Lonie, S. Kosakovsky Pond, W. Maier, A. Nekrutenko, J. Taylor and S. Weaver, No more business as usual: Agile and effective responses to emerging pathogen threats require open data and open analytics., *PLoS Pathogens* **16**(8) (2020), e1008643. doi:10.1371/journal.ppat.1008643.

[65] P.A. Ewels, A. Peltzer, S. Fillinger, H. Patel, J. Alneberg, A. Wilm, M.U. Garcia, P. Di Tommaso and S. Nahnsen, The nf-core framework for community-curated bioinformatics pipelines., *Nature Biotechnology* **38**(3) (2020), 276–278. doi:10.1038/s41587-020-0439-x.

[66] P. Di Tommaso, M. Chatzou, E.W. Floden, P.P. Barja, E. Palumbo and C. Notredame, Nextflow enables reproducible computational workflows., *Nature Biotechnology* **35**(4) (2017), 316–319. doi:10.1038/nbt.3820.

[67] J. Köster and S. Rahmann, Snakemake–a scalable bioinformatics workflow engine., *Bioinformatics* **28**(19) (2012), 2520–2522. doi:10.1093/bioinformatics/bts480.

[68] F. Bietrix, J.M. Carazo, S. Capella-Gutierrez, F. Coppens, M.L. Chiusano, R. David, J.M. Fernandez, M. Fratelli, J.-K. Heriche, C. Goble, P. Gribbon, P. Holub, R. P. Joosten, S. Leo, S. Owen, H. Parkinson, R. Pieruschka, L. Pireddu, L. Porcu, M. Raess, L. Rodriguez- Navas, A. Scherer, S. Soiland-Reyes and J. Tang, EOSC-Life Methodology framework to enhance reproducibility within EOSC-Life, *Zenodo* (2021). doi:10.5281/zenodo.4705078.

[69] M.R. Crusoe, S. Abeln, A. Iosup, P. Amstutz, J. Chilton, N. Tijanić, H. Ménager, S. Soiland-Reyes and C. Goble, Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language, *Communications of the ACM* (2021), Accepted. doi:10.1145/3486897.

[70] C. Goble, S. Soiland-Reyes, F. Bacall, S. Owen, A. Williams, I. Eguinoa, B. Droesbeke, S. Leo, L. Pireddu, L. Rodríguez-Navas, J.M. Fernández, S. Capella-Gutierrez, H. Ménager, B. Grüning, B. Serrano-Solano, P. Ewels and F. Coppens, Implementing FAIR Digital Objects in the EOSC-Life Workflow Collaboratory, *Zenodo* (2021). doi:10.5281/zenodo.4605654.

[71] S. Soiland-Reyes, P. Alper and C. Goble, Tracking Workflow Execution With TavernaPROV, in: *ProvenanceWeek 2016*, 2016, PROV: Three Years Later. doi:10.5281/zenodo.51314.

[72] S. Soiland-Reyes, M. Gamble and R. Haines, *Research Object Bundle 1.0*, 2014, See https://w3id.org/bundle/2014-11-05/. doi:10.5281/zenodo.12586. https://w3id.org/bundle/2014-11-05/.

[73] S. Walton, L. Livermore, O. Bánki, R. Cubey, R. Drinkwater, M. Englund, C. Goble, Q. Groom, C. Kermorvant, I. Rey, C. Santos, B. Scott, A. Williams and Z. Wu, Landscape analysis for the Specimen Data Refinery, *Research Ideas and Outcomes* **6** (2020). doi:10.3897/rio.6.e57602.

[74] K. De Smedt, D. Koureas and P. Wittenburg, FAIR digital objects for science: from data pieces to actionable knowledge units, *Publications* **8**(2) (2020), 21. doi:10.3390/publications8020021.

[75] D. Lowe and G. Bayarri, Protein Ligand Complex MD Setup tutorial using BioExcel Building Blocks (biobb) (jupyter notebook), 2021. doi:10.48546/workflowhub.workflow.56.1.

[76] J. Zhao, J.M. Gomez-Perez, K. Belhajjame, G. Klyne, E. Garcia-Cuesta, A. Garrido, K. Hettne, M. Roos, D. De Roure and C. Goble, Why workflows break — Understanding and combating decay in Taverna workflows, in: *2012 IEEE 8th International Conference on E-Science*, IEEE, 2012, pp. 1–9. ISBN 978-1-4673-4466-1. doi:10.1109/eScience.2012.6404482. https://www.research.manchester.ac.uk/portal/files/174861334/why_decay.pdf.

[77] G. Alterovitz, D. Dean, C. Goble, M.R. Crusoe, S. Soiland-Reyes, A. Bell, A. Hayes, A. Suresh, A. Purkayastha, C.H. King, D. Taylor, E. Johanson, E.E. Thompson, E. Donaldson, H. Morizono, H. Tsang, J.K. Vora, J. Goecks, J. Yao, J.S. Almeida, J. Keeney, K. Addepalli, K. Krampis, K.M. Smith, L. Guo, M. Walderhaug, M. Schito, M. Ezewudo, N. Guimera, P. Walsh, R. Kahsay, S. Gottipati, T.C. Rodwell, T. Bloom, Y. Lai, V. Simonyan and R. Mazumder, Enabling precision medicine via standard communication of HTS provenance, analysis, and results., *PLoS Biology* **16**(12) (2018), e3000099. doi:10.1371/journal.pbio.3000099.

[78] IEEE Standard for Bioinformatics Analyses Generated by High-Throughput Sequencing (HTS) to Facilitate Communication, IEEE Std 2791-2020. ISBN 978-1-5044-6466-6. doi:10.1109/IEEESTD.2020.9094416.

[79] S. Soiland-Reyes, Describing and packaging workflows using RO-Crate and BioCompute Objects, Zenodo, 2021, Webinar for U.S. Food and Drug Administration (FDA), 2021-05-12. doi:10.5281/zenodo.4633732.

[80] Managing large files - GitHub Docs. https://docs.github.com/en/repositories/working-with-files/managing-large-files.

[81] K. Chard, M. D'Arcy, B. Heavner, I. Foster, C. Kesselman, R. Madduri, A. Rodriguez, S. Soiland-Reyes, C. Goble, K. Clark, E.W. Deutsch, I. Dinov, N. Price and A. Toga, I'll take that to go: Big data bags and minimal identifiers for exchange of large, complex datasets, in: *2016 IEEE International Conference on Big Data (Big Data)*, IEEE, 2016, pp. 319–328. ISBN 978-1-4673-9005-7. doi:10.1109/BigData.2016.7840618. https://static.aminer.org/pdf/fa/bigdata2016/BigD418.pdf.

[82] N. Thieberger and L. Barwick, Keeping records of language diversity in Melanesia: The Pacific and Regional Archive for Digital Sources in Endangered Cultures (PARADISEC), in: *Melanesian Languages on the Edge of Asia: Challenges for the 21st Century*, N. Evans and M. Klamer, eds, Language Documentation & Conservation Special Publication, Vol. SP05, University of Hawai'i Press, 2012, pp. 239–253. ISBN 978-0-9856211-2-4. doi:10125/4567.

[83] T. Miksa, S. Simms, D. Mietchen and S. Jones, Ten principles for machine-actionable data management plans, *PLoS Computational Biology* **15**(3) (2019), e1006750. doi:10.1371/journal.pcbi.1006750.

[84] J. Cardoso, D. Proença and J. Borbinha, Machine-Actionable Data Management Plans: A Knowledge Retrieval Approach to Automate the Assessment of Funders' Requirements, in: *Advances in Information Retrieval*, J.M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M.J. Silva and F. Martins, eds, Springer International Publishing, Cham, 2020, pp. 118–125. ISBN 978-3-030-45442-5. doi:10.1007/978-3-030-45442-5_15.

[85] P. Walk, T. Miksa and P. Neish, RDA DMP Common Standard for Machine-actionable Data Management Plans, *Research Data Alliance* (2019). doi:10.15497/rda00039.

[86] J. Cardoso, L.J. Garcia Castro, F. Ekaputra, M.-C. Jacquemot-Perbal, T. Miksa and J. Borbinha, Towards semantic representation of machine-actionable Data Management Plans, *PUBLISSO* (2020). doi:10.4126/frl01-006423289. https://repository.publisso.de/resource/frl:6423289.

[87] R. Albertoni, D. Browning, S. Cox, A. Gonzalez Beltran, A. Perego, P. Winstanley and Dataset Exchange Working Group, Data Catalog Vocabulary (DCAT) - Version 2, W3C Recommendation, 2020. https://www.w3.org/TR/2020/REC-vocab-dcat-2-20200204/.

[88] R.L. Grossman, A. Heath, M. Murphy, M. Patterson and W. Wells, A case for data commons: toward data science as a service, *Computing in science & engineering* **18**(5) (2016), 10–20. doi:10.1109/MCSE.2016.92.

[89] M. Barker, R. Wilkinson and A. Treloar, The Australian Research Data Commons, *Data Science Journal* **18** (2019). doi:10.5334/dsj-2019-044.

[90] M.A. Jensen, V. Ferretti, R.L. Grossman and L.M. Staudt, The NCI Genomic Data Commons as an engine for precision medicine., *Blood* **130**(4) (2017), 453–459. doi:10.1182/blood-2017-03-735654.

[91] M. Crosas, Harvard Data Commons, 2020, European Dataverse Workshop 2020, Tromsø, Norway. 2020-01-23/–24. ISSN 2387-3086. doi:10.7557/5.5422.

[92] M. Crosas, The Dataverse Network: An Open-Source Application for Sharing, Discovering and Preserving Data, *D-Lib Magazine* **17**(1/2) (2011). doi:10.1045/january2011-crosas.

[93] K. Chard, S. Tuecke and I. Foster, Efficient and Secure Transfer, Synchronization, and Sharing of Big Data, *IEEE Cloud Computing* **1**(3) (2014), 46–55. doi:10.1109/MCC.2014.52.

[94] L. Koesten, P. Vougiouklis, E. Simperl and P. Groth, Dataset reuse: toward translating principles to practice, *Patterns (New York, N.Y.)* **1**(8) (2020), 100136. doi:10.1016/j.patter.2020.100136.

[95] J. Leipzig, D. Nüst, C.T. Hoyt, K. Ram and J. Greenberg, The role of metadata in reproducible computational research, *Patterns* **2**(9) (2021), 100322. doi:10.1016/j.patter.2021.100322.

[96] J.F. Claerbout and M. Karrenbach, Electronic documents give reproducible research a new meaning, in: *SEG Technical Program Expanded Abstracts 1992*, Society of Exploration Geophysicists, 1992, pp. 601–604. doi:10.1190/1.1822162.

[97] H. Van de Sompel and C. Lagoze, Interoperability for the Discovery, Use, and Re-Use of Units of Scholarly Communication, *CTWatch Quarterly* **3**(3) (2007). http://icl.utk.edu/ctwatch/quarterly/articles/2007/08/interoperability-for-the-discovery-use-and-re-use-of-units-of-scholarly-communication/.

[98] C.A. Goble, J. Bhagat, S. Aleksejevs, D. Cruickshank, D. Michaelides, D. Newman, M. Borkum, S. Bechhofer, M. Roos, P. Li and D. De Roure, myExperiment: a repository and social network for the sharing of bioinformatics workflows., *Nucleic Acids Research* **38**(Web Server issue) (2010), W677–82. doi:10.1093/nar/gkq429.

[99] D. Newman, S. Bechhofer and D.D. Roure, myExperiment: An ontology for e-Research, in: *Proceedings of the Workshop on Semantic Web Applications in Scientific Discourse (SWASD 2009)*, T. Clark, J.S. Luciano, M.S. Marshall, E. Prud'Hommeaux and S. Stephens, eds, CEUR Workshop Proceedings, Vol. 523, CEUR-WS, 2009, 2009-10-25. ISSN 1613-0073. http://ceur-ws.org/Vol-523/Newman.pdf.

[100] myExperiment Ontology Modules, 2009. http://web.archive.org/web/20091115080336/http%3a%2f%2frdf.myexperiment.org/ontologies.

[101] P. Ciccarese, R. Sanderson and B. Young, Web Annotation Data Model, W3C Recommendation, W3C, 2017, https://www.w3.org/TR/2017/REC-annotation-model-20170223/.

[102] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik and J. Zhao, PROV-O: The PROV Ontology, Technical Report, 2013, W3C Recommendation Note 30 April 2013. http://www.w3.org/TR/2013/REC-prov-o-20130430/.

[103] M. Katsumi and M. Grüninger, What Is Ontology Reuse?, in: *Formal Ontology in Information Systems*, R. Ferrario and W. Kuhn, eds, Frontiers in Artificial Intelligence and Applications, Vol. 283, IOS Press, 2016, pp. 9–22. ISBN 978-1-61499-660-6. doi:10.3233/978-1-61499-660-6-9.

[104] A. Garcia-Silva, J.M. Gomez-Perez, R. Palma, M. Krystek, S. Mantovani, F. Foglini, V. Grande, F. De Leo, S. Salvi, E. Trasatti, V. Romaniello, M. Albani, C. Silvagni, R. Leone, F. Marelli, S. Albani, M. Lazzarini, H.J. Napier, H.M. Glaves, T. Aldridge, C. Meertens, F. Boler, H.W. Loescher, C. Laney, M.A. Genazzio, D. Crawl and I. Altintas, Enabling FAIR research in Earth Science through research objects, *Future Generation Computer Systems* **98** (2019), 550–564. doi:10.1016/j.future.2019.03.046.

[105] C. Kyle, G. Niall, H. Mihael, K. Kacper, L. Bertram, M. Timothy, N. Jarek, S. Victoria, T. Ian, T. Thomas and et al., Toward Enabling Reproducibility for Data-Intensive Research Using the Whole Tale Platform, *Advances in Parallel Computing* **36**(Parallel Computing: Technology Trends) (2020), 766–778–. doi:10.3233/APC200107.

[106] K. Chard, N. Gaffney, M.B. Jones, K. Kowalik, B. Ludascher, T. McPhillips, J. Nabrzyski, V. Stodden, I. Taylor, T. Thelen, M.J. Turk and C. Willis, Application of BagIt-Serialized Research Object Bundles for Packaging and Re-Execution of Computational Analyses, in: *15th International Conference on eScience (eScience 2019)*, IEEE, 2019, pp. 514–521. ISBN 978-1-7281-2451-3. doi:10.1109/eScience.2019.00068.

[107] D. Foundation, Digital Object Interface Protocol Specification, version 2.0, Technical Report, 2018. https://www.dona.net/sites/default/files/2018-11/DOIPv2Spec_1.pdf.

[108] S. Cohen-Boulakia, K. Belhajjame, O. Collin, J. Chopard, C. Froidevaux, A. Gaignard, K. Hinsen, P. Larmande, Y.L. Bras, F. Lemoine, F. Mareuil, H. Ménager, C. Pradal and C. Blanchet, Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities, *Future Generation Computer Systems* **75** (2017), 284–298. doi:10.1016/j.future.2017.01.012.

[109] J. Kim, E. Deelman, Y. Gil, G. Mehta and V. Ratnakar, Provenance trails in the Wings/Pegasus system, *Concurrency and Computation: Practice and Experience* **20**(5) (2008), 587–597. doi:10.1002/cpe.1228.

[110] B. Grüning, J. Chilton, J. Köster, R. Dale, N. Soranzo, M. van den Beek, J. Goecks, R. Backofen, A. Nekrutenko and J. Taylor, Practical computational reproducibility in the life sciences., *Cell Systems* **6**(6) (2018), 631–635. doi:10.1016/j.cels.2018.03.014.

[111] A. Almeida, A.L. Mitchell, M. Boland, S.C. Forster, G.B. Gloor, A. Tarkowska, T.D. Lawley and R.D. Finn, A new genomic blueprint of the human gut microbiota., *Nature* **568**(7753) (2019), 499–504. doi:10.1038/s41586-019-0965-1.

[112] EMBL-EBI Microbiome Informatics Team, FTP index of /pub/databases/metagenomics/umgs_analyses/, 2019. http://ftp.ebi.ac.uk/pub/databases/metagenomics/umgs_analyses/.

[113] EMBL-EBI Microbiome Informatics Team, GitHub - Finn-Lab/MGS-gut: Analysing Metagenomic Species (MGS). https://github.com/Finn-Lab/MGS-gut.

[114] S. Soiland-Reyes, I am looking for which bioinformatics journals encourage authors to submit their code/pipeline/workflow supporting data analysis, 2020. https://twitter.com/soilandreyes/status/1250721245622079488.

[115] Giving software its due, *Nature Methods* **16**(3) (2019), 207–207. doi:10.1038/s41592-019-0350-x.

[116] F.C.Y. Benureau and N.P. Rougier, Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions., *Frontiers in Neuroinformatics* **11** (2017), 69. doi:10.3389/fninf.2017.00069.

[117] C. Goble, What is Reproducibility? The R* Brouhaha, Hannover, Germany, 2016. http://repscience2016.research-infrastructures.eu/img/CaroleGoble-ReproScience2016v2.pdf.

[118] S. Möller, S.W. Prescott, L. Wirzenius, P. Reinholdtsen, B. Chapman, P. Prins, S. Soiland-Reyes, F. Klötzl, A. Bagnacani, M. Kalaš, A. Tille and M.R. Crusoe, Robust Cross-Platform Workflows: How Technical and Scientific Communities Collaborate to Develop, Test and Share Best Practices for Data Analysis, *Data Science and Engineering* **2**(3) (2017), 232–244. doi:10.1007/s41019-017-0050-4.

[119] B. Grüning, R. Dale, A. Sjödin, B.A. Chapman, J. Rowe, C.H. Tomkins-Tinch, R. Valieris, J. Köster and B. Team, Bioconda: sustainable and comprehensive software distribution for the life sciences., *Nature Methods* **15**(7) (2018), 475–476. doi:10.1038/s41592-018-0046-7.

[120] F. da Veiga Leprevost, B.A. Grüning, S. Alves Aflitos, H.L. Röst, J. Uszkoreit, H. Barsnes, M. Vaudel, P. Moreno, L. Gatto, J. Weber, M. Bai, R.C. Jimenez, T. Sachsenberg, J. Pfeuffer, R. Vera Alvarez, J. Griss, A.I. Nesvizhskii and Y. Perez-Riverol, BioContainers: an open-source and community-driven framework for software standardization., *Bioinformatics* **33**(16) (2017), 2580–2582. doi:10.1093/bioinformatics/btx192.

[121] S. Möller, H.N. Krabbenhöft, A. Tille, D. Paleino, A. Williams, K. Wolstencroft, C. Goble, R. Holland, D. Belhachemi and C. Plessy, Community-driven computational biology with Debian Linux., *BMC Bioinformatics* **11 Suppl 12** (2010), S5. doi:10.1186/1471-2105-11-S12-S5.

[122] E. Afgan, D. Baker, B. Batut, M. van den Beek, D. Bouvier, M. Cech, J. Chilton, D. Clements, N. Coraor, B.A. Grüning, A. Guerler, J. Hillman-Jackson, S. Hiltemann, V. Jalili, H. Rasche, N. Soranzo, J. Goecks, J. Taylor, A. Nekrutenko and D. Blankenberg, The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update., *Nucleic Acids Research* **46**(W1) (2018), W537–W544. doi:10.1093/nar/gky379.

[123] R. Troncy, W. Bailer, M. Höffernig and M. Hausenblas, VAMP: a service for validating MPEG-7 descriptions w.r.t. to formal profile definitions, *Multimedia tools and applications* **46**(2–3) (2010), 307–329, (Available from institutional repository https://www.persistent-identifier.nl/urn:nbn:nl:ui:18-14511). doi:10.1007/s11042-009-0397-2.

[124] D. Agarwal, C. Goble, S. Soiland-Reyes, U. Sarkans, D. Noesgaard, U. Schindler, M. Fenner, P. Manghi, S. Stall, C. Coward and C. Erdmann, Data Citation Community of Practice - 8 June 2021 Workshop, Zenodo/AGU, 2021. doi:10.5281/zenodo.4916734. https://data.agu.org/DataCitationCoP/2nd-workshop-data-citation.

[125] M.B. Jones, S. Richard, D. Vieglais, A. Shepherd, R. Duerr, D. Fils and L. McGibbney, Science-on-Schema.org v1.2.0, Technical Report, 2021. doi:10.5281/zenodo.4477164. https://science-on-schema.org/.

[126] A. Brand, L. Allen, M. Altman, M. Hlava and J. Scott, Beyond authorship: attribution, contribution, collaboration, and credit, *Learned Publishing* **28**(2) (2015), 151–155. doi:10.1087/20150211.

[127] RDF Working Group, RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation, 2014. https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

Appendix A. Formalizing RO-Crate in First Order Logic

Below is a formalization of the concept of RO-Crate as a set of relations using First Order Logic:

A.1. Language

Definition of language $\mathcal{L}_{rocrate}$:

$$\begin{aligned}
\mathcal{L}_{rocrate} &= \{Property(p), Class(c), Value(x), \mathbb{R}, \mathbb{S}\} \\
\mathbb{D} &= \mathbb{IRI} \\
\mathbb{IRI} &\equiv \text{IRIs as defined in RFC3987} \\
\mathbb{R} &\equiv \text{real or integer numbers} \\
\mathbb{S} &\equiv \text{literal strings}
\end{aligned}$$

The domain of discourse $\mathbb{D}$ is the set of $\mathbb{IRI}$ identifiers [30] (notation `<http://example.com/>`)[110], with additional descriptions using numbers $\mathbb{R}$ (notation 13.37) and literal strings $\mathbb{S}$ (notation "Hello").

From this formalised language $\mathcal{L}_{rocrate}$ we can interpret an RO-Crate in any representation that can gather these descriptions, their properties, classes, and literal attributes.

A.2. Minimal RO-Crate

Below we use $\mathcal{L}_{rocrate}$ to define a minimal[111] RO-Crate:

$$\begin{aligned}
ROCrate(R) \vDash\ & Root(R) \land Mentions(R,R) \land hasPart(R,d) \land \\
& Mentions(R,d) \land DataEntity(d) \land \\
& Mentions(R,c) \land ContextualEntity(c) \\
\forall r\ Root(r) \Rightarrow\ & Dataset(r) \land name(r,n) \land description(r,d) \land \\
& datePublished(r,date) \land license(e,l) \\
\forall e \forall n\ name(e,n) \Rightarrow\ & Value(n) \\
\forall e \forall s\ description(e,s) \Rightarrow\ & Value(s) \\
\forall e \forall d\ datePublished(e,d) \Rightarrow\ & Value(d) \\
\forall e \forall l\ license(e,l) \Rightarrow\ & ContextualEntity(l) \\
DataEntity(e) \equiv\ & File(e) \oplus Dataset(e)
\end{aligned}$$

[110] For simplicity, blank nodes are not included in this formalisation, as RO-Crate recommends the use of IRI identifiers: https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#describing-entities-in-json-ld

[111] The full list of types, relations and attribute properties from the RO-Crate specification are not included. Examples shown include $datePublished$, $CreativeWork$ and $name$.

$$\begin{aligned}
Entity(e) &\equiv DataEntity(e) \vee ContextualEntity(e) \\
\forall e\, Entity(e) &\Rightarrow type(e,c) \wedge Class(c) \\
\forall e\, ContextualEntity(e) &\Rightarrow name(e,n) \\
Mentions(R,s) &\vDash Relation(s,p,e) \oplus Attribute(s,p,l) \\
Relation(s,p,o) &\vDash Entity(s) \wedge Property(p) \wedge Entity(o) \\
Attribute(s,p,v) &\vDash Entity(s) \wedge Property(p) \wedge Value(v) \\
Value(v) &\equiv v \in \mathbb{R} \oplus v \in \mathbb{S}
\end{aligned}$$

An $ROCrate(R)$ is defined as a self-described *Root Data Entity*, which describes and contains parts (*data entities*), which are further described in *contextual entities*. These terms align with their use in the RO-Crate 1.1 terminology[112].

The $Root(r)$ is a type of $Dataset(r)$, and must have the metadata to literal attributes to provide a $name$, $description$ and $datePublished$, as well as a contextual entity identifying its license. These predicates correspond to the RO-Crate 1.1 requirements for the root data entity[113].

The concept of an $Entity(e)$ is introduced as being either a $DataEntity(e)$, a $ContextualEntity(e)$, or both[114]. Any $Entity(e)$ must be typed with at least one $Class(c)$, and every $ContextualEntity(e)$ must also have a $name(e,n)$; this corresponds to expectations for any *referenced contextual entity* (section 2.2.3).

For simplicity in this formalization (and to assist production rules below) R is a constant representing a single RO-Crate, typically written to independent RO-Crate Metadata files. R is used by $Mentions(R,e)$ to indicate that e is an Entity described by the RO-Crate and therefore its metadata (a set of $Relation$ and $Attribute$ predicates) form part of the RO-Crate serialization. $Relation(s,p,o)$ and $Attribute(s,p,x)$ are defined as a *subject—predicate—object* triple pattern from an $Entity(s)$ using a $Property(p)$ to either another $Entity(o)$ or a $Value(x)$ value.

A.3. Example of formalized RO-Crate

The below is an example RO-Crate represented using the above formalisation, assuming a base URI of `<http://example.com/ro/123/>`:

$ROCrate$(`<http://example.com/ro/123/>`)

$name$(`<http://example.com/ro/123/>`,

"Data files associated with the manuscript:Effects of ...")

$description$(`<http://example.com/ro/123/`,

"Palliative care planning for nursing home residents ...")

$license$(`<http://example.com/ro/123/>`,

[112] https://www.researchobject.org/ro-crate/1.1/terminology
[113] https://www.researchobject.org/ro-crate/1.1/root-data-entity.html#direct-properties-of-the-root-data-entity
[114] https://www.researchobject.org/ro-crate/1.1/contextual-entities.html#contextual-vs-data-entities

    `<https://spdx.org/licenses/CC-BY-4.0>`)

datePublished(`<http://example.com/ro/123/>`, “2017-02-23”)

hasPart(`<http://example.com/ro/123/>`, `<http://example.com/ro/123/file.txt>`)

hasPart(`<http://example.com/ro/123/>`, `<http://example.com/ro/123/interviews/>`)

ContextualEntity(`<https://spdx.org/licenses/CC-BY-4.0>`)

name(`<https://spdx.org/licenses/CC-BY-4.0>`,

    Creative Commons Attribution 4.0”)

ContextualEntity(`<https://spdx.org/licenses/CC-BY-NC-4.0>`)

name(`<https://spdx.org/licenses/CC-BY-NC-4.0>`,

    Creative Commons Attribution Non Commercial 4.0”)

File(`<http://example.com/ro/123/survey.csv>`)

name(`<http://example.com/ro/123/survey.csv>`, “Survey of care providers”)

Dataset(`<http://example.com/ro/123/interviews/>`)

name(`<http://example.com/ro/123/interviews/>`,

    “Audio recordings of care provider interviews”)

license(`<http://example.com/ro/123/interviews/>`,

    `<https://spdx.org/licenses/CC-BY-NC-4.0>`)

Notable from this triple-like formalization is that a RO-Crate R is fully represented as a tree at depth 2 helped by the use of Iri nodes. For instance the aggregation from the root entity *hasPart*(…`interviews/>`) is at same level as the data entity’s property *license*(…`CC-BY-NC-4.0>`) and that contextual entity’s attribute *name*(…Non Commercial 4.0”). As shown in section 2.3.1, the RO-Crate Metadata File serialization is an equivalent shallow tree, although at depth 3 to cater for the JSON-LD preamble of `"@context"` and `"@graph"`.

In reality many additional attributes and contextual types from schema.org types like http://schema.org/affiliation and http://schema.org/Organization would be used to further describe the RO-Crate and its entities, but as these are optional (*SHOULD* requirements) they do not form part of this formalization.

A.4. Mapping to RDF with schema.org

A formalized RO-Crate in $\mathcal{L}_{rocrate}$ can be mapped to different serializations. Assume a simplified[115] language $\mathcal{L}_{RDF}$ based on the RDF abstract syntax [127]:

$$
\begin{aligned}
\mathcal{L}_{RDF} &\equiv \{Triple(s,p,o), IRI(i), BlankNode(b), Literal(s), \mathbb{IRI}, \mathbb{S}, \mathbb{R}\} \\
\mathbb{D}_{RDF} &\equiv \mathbb{S} \\
\forall i\, IRI(i) &\Rightarrow i \in \mathbb{IRI} \\
\forall s \forall p \forall o\, Triple(s,p,o) &\Rightarrow \big(IRI(s) \lor BlankNode(s)\big) \land \\
&\quad IRI(p) \land \\
&\quad \big(IRI(o) \lor BlankNode(o) \lor Literal(o)\big) \\
Literal(v) &\vDash Value(v) \land Datatype(v,t) \land IRI(t) \\
\forall v\, Value(v) &\Rightarrow v \in \mathbb{S} \\
LanguageTag(v,l) &\equiv Datatype\big(v, \\
&\quad \texttt{<http://www.w3.org/1999/02/22-rdf-syntax-ns\#langString>}\big)
\end{aligned}
$$

Below follows a mapping from $\mathcal{L}_{rocrate}$ to $\mathcal{L}_{RDF}$ using schema.org as vocabulary:

$$
\begin{aligned}
Property(p) &\Rightarrow type(p, \texttt{<http://www.w3.org/2000/01/rdf-schema\#Property>}) \\
Class(c) &\Rightarrow type(c, \texttt{<http://www.w3.org/2000/01/rdf-schema\#Class>}) \\
Dataset(d) &\Rightarrow type(d, \texttt{<http://schema.org/Dataset>}) \\
File(f) &\Rightarrow type(f, \texttt{<http://schema.org/MediaObject>}) \\
ContextualEntity(e) &\Rightarrow type(f, \texttt{<http://schema.org/Thing>}) \\
CreativeWork(e) &\Rightarrow ContextualEntity(e) \land type(e, \texttt{<http://schema.org/CreativeWork>}) \\
hasPart(e,t) &\Rightarrow Relation(e, \texttt{<http://schema.org/hasPart>}, t) \\
name(e,n) &\Rightarrow Attribute(e, \texttt{<http://schema.org/name>}, n) \\
description(e,s) &\Rightarrow Attribute(e, \texttt{<http://schema.org/description>}, s) \\
datePublished(e,d) &\Rightarrow Attribute(e, \texttt{<http://schema.org/datePublished>}, d) \\
license(e,l) &\Rightarrow Relation(e, \texttt{<http://schema.org/license>}, l) \land CreativeWork(l) \\
type(e,t) &\Rightarrow Relation(e, \texttt{<http://www.w3.org/1999/02/22-rdf-syntax-ns\#type>}, t) \\
&\quad \land Class(t)
\end{aligned}
$$

[115]This simplification does not cover the extensive list of literal datatypes built-in to RDF 1.1, only strings and decimal real numbers. Likewise, language of literals are not included.

$$
\begin{aligned}
String(s) &\equiv Value(s) \wedge s \in \mathbb{S} \\
String(s) &\Rightarrow Datatype(s, \texttt{<http://www.w3.org/2001/XMLSchema\#string>}) \\
Decimal(d) &\equiv Value(d) \wedge d \in \mathbb{R} \\
Decimal(d) &\Rightarrow Datatype(d, \texttt{<http://www.w3.org/2001/XMLSchema\#decimal>}) \\
Relation(s, p, o) &\Rightarrow Triple(s, p, o) \wedge IRI(s) \wedge IRI(o) \\
Attribute(s, p, o) &\Rightarrow Triple(s, p, o) \wedge IRI(s) \wedge Literal(o)
\end{aligned}
$$

Note that in the JSON-LD serialization of RO-Crate, the expression of $Class$ and $Property$ is typically indirect: The JSON-LD `@context` maps to schema.org IRIs, which, when resolved as Linked Data, embed their formal definition as RDFa. Extensions may however include such term definitions directly in the RO-Crate.

A.5. RO-Crate 1.1 Metadata File Descriptor

An important RO-Crate principle is that of being **self-described** Therefore the serialisation of the RO-Crate into a file should also describe itself in a Metadata File Descriptor[116], indicating it is *about* (describing) the RO-Crate root data entity, and that it $conformsTo$ a particular version of the RO-Crate specification:

$$
\begin{aligned}
about(s, o) &\Rightarrow Relation(s, \texttt{<http://schema.org/about>}, o) \\
conformsTo(s, o) &\Rightarrow Relation(s, \texttt{<http://purl.org/dc/terms/conformsTo>}, o) \\
MetadataFile(m) &\Rightarrow CreativeWork(m) \wedge about(m, R) \wedge ROCrate(R) \wedge \\
&\qquad conformsTo(m, \texttt{<https://w3id.org/ro/crate/1.1>})
\end{aligned}
$$

Note that although the metadata file necessarily is an *information resource* written to disk or served over the network (as JSON-LD), it is not considered to be a contained *part* of the RO-Crate in the form of a *data entity*, rather it is described only as a *contextual entity*.

In the conceptual model the *RO-Crate Metadata File* can be seen as the top-level node that describes the *RO-Crate Root*, however in the formal model (and the JSON-LD format) the metadata file descriptor is an additional contextual entity that is not affecting the depth-limit of the RO-Crate.

A.6. Forward-chained Production Rules for JSON-LD

Combining the above predicates and schema.org mapping with rudimentary JSON templates, these forward-chaining production rules can output JSON-LD according to the RO-Crate 1.1 specification[117]:

[116] https://www.researchobject.org/ro-crate/1.1/root-data-entity.html#ro-crate-metadata-file-descriptor

[117] **Limitations:** Contextual entities not related from the RO-Crate (e.g. using inverse relations to a data entity) would not be covered by the single direction $Mentions(R, s)$ production rule; see GitHub issue ResearchObject/ro-crate#122. The $datePublished(e, d)$ rule do not include syntax checks for the ISO 8601 datetime format. Compared with RO-Crate

$$
\begin{aligned}
Mentions(R, s) \land Relation(s, p, o) &\Rightarrow Mentions(R, o)\\
IRI(i) &\Rightarrow \texttt{"}i\texttt{"}\\
Decimal(d) &\Rightarrow d\\
String(s) &\Rightarrow \texttt{"}s\texttt{"}\\
\forall e \forall t\ type(e, t) &\Rightarrow \{\texttt{"@id"}:e,\\
&\quad \texttt{"@type"}:t\\
&\}\\
\forall s \forall p \forall o\ Relation(s, p, o) &\Rightarrow \{\texttt{"@id"}:s,\\
&\quad p\texttt{: \{ "@id"}:o\}\\
&\}\\
\forall s \forall p \forall v\ Attribute(s, p, v) &\Rightarrow \{\texttt{"@id"}:s,\\
&\quad p:v\\
&\}\\
\forall r \forall c ROCrate(r) &\Rightarrow \texttt{\{ "@graph": [}\\
&\quad\ Mentions(r, c) *\\
&\quad \texttt{]}\\
&\}\\
R &\equiv \texttt{<./>}\\
R &\Rightarrow MetadataFile(\texttt{<ro-crate-metadata.json>})
\end{aligned}
$$

This exposes the first order logic domain of discourse of IRIs, with rational numbers and strings as their corresponding JSON-LD representation. These production rules first grow the graph of R by adding a transitive rule – anything described in R which is related to o, means that o is also mentioned by the $ROCrate(R)$. For simplicity this rule is one-way; in theory the graph can also contain free-standing contextual entities that have outgoing relations to data- and contextual entities, but these are proposed to be bound to the root data entity with schema.org relation .

examples, this generated JSON-LD does not use a $@context$ as the IRIs are produced unshortened, a post-step could do JSON-LD Flattening with a versioned RO-Crate context. The `@type` expansion is included for clarity, even though this is also implied by the $type(e, t)$ expansion to $Relation(e, \texttt{xsd:type})$.

Appendix B. RO-Crate Community

As of 2021-10-04, the *RO-Crate* Community members are:

- Peter Sefton https://orcid.org/0000-0002-3545-944X (co-chair)
- Stian Soiland-Reyes https://orcid.org/0000-0001-9842-9718 (co-chair)
- Eoghan Ó Carragáin https://orcid.org/0000-0001-8131-2150 (emeritus chair)
- Oscar Corcho https://orcid.org/0000-0002-9260-0753
- Daniel Garijo https://orcid.org/0000-0003-0454-7145
- Raul Palma https://orcid.org/0000-0003-4289-4922
- Frederik Coppens https://orcid.org/0000-0001-6565-5145
- Carole Goble https://orcid.org/0000-0003-1219-2137
- José María Fernández https://orcid.org/0000-0002-4806-5140
- Kyle Chard https://orcid.org/0000-0002-7370-4805
- Jose Manuel Gomez-Perez https://orcid.org/0000-0002-5491-6431
- Michael R Crusoe https://orcid.org/0000-0002-2961-9670
- Ignacio Eguinoa https://orcid.org/0000-0002-6190-122X
- Nick Juty https://orcid.org/0000-0002-2036-8350
- Kristi Holmes https://orcid.org/0000-0001-8420-5254
- Jason A. Clark https://orcid.org/0000-0002-3588-6257
- Salvador Capella-Gutierrez https://orcid.org/0000-0002-0309-604X
- Alasdair J. G. Gray https://orcid.org/0000-0002-5711-4872
- Stuart Owen https://orcid.org/0000-0003-2130-0865
- Alan R Williams https://orcid.org/0000-0003-3156-2105
- Giacomo Tartari https://orcid.org/0000-0003-1130-2154
- Finn Bacall https://orcid.org/0000-0002-0048-3300
- Thomas Thelen https://orcid.org/0000-0002-1756-2128
- Hervé Ménager https://orcid.org/0000-0002-7552-1009
- Laura Rodríguez-Navas https://orcid.org/0000-0003-4929-1219
- Paul Walk https://orcid.org/0000-0003-1541-5631
- brandon whitehead https://orcid.org/0000-0002-0337-8610
- Mark Wilkinson https://orcid.org/0000-0001-6960-357X
- Paul Groth https://orcid.org/0000-0003-0183-6910
- Erich Bremer https://orcid.org/0000-0003-0223-1059
- LJ Garcia Castro https://orcid.org/0000-0003-3986-0510
- Karl Sebby https://orcid.org/0000-0001-6022-9825
- Alexander Kanitz https://orcid.org/0000-0002-3468-0652
- Ana Trisovic https://orcid.org/0000-0003-1991-0533
- Gavin Kennedy https://orcid.org/0000-0003-3910-0474
- Mark Graves https://orcid.org/0000-0003-3486-8193
- Jasper Koehorst https://orcid.org/0000-0001-8172-8981
- Simone Leo https://orcid.org/0000-0001-8271-5429
- Marc Portier https://orcid.org/0000-0002-9648-6484
- Paul Brack https://orcid.org/0000-0002-5432-2748
- Milan Ojsteršek https://orcid.org/0000-0003-1743-8300
- Bert Droesbeke https://orcid.org/0000-0003-0522-5674

- Chenxu Niu https://github.com/UstcChenxu
- Kosuke Tanabe https://orcid.org/0000-0002-9986-7223
- Tomasz Miksa https://orcid.org/0000-0002-4929-7875
- Marco La Rosa https://orcid.org/0000-0001-5383-6993
- Cedric Decruw https://orcid.org/0000-0001-6387-5988
- Andreas Czerniak https://orcid.org/0000-0003-3883-4169
- Jeremy Jay https://orcid.org/0000-0002-5761-7533
- Sergio Serra https://orcid.org/0000-0002-0792-8157
- Ronald Siebes https://orcid.org/0000-0001-8772-7904
- Shaun de Witt https://orcid.org/0000-0003-4196-3658
- Shady El Damaty https://orcid.org/0000-0002-2318-4477
- Douglas Lowe https://orcid.org/0000-0002-1248-3594
- Sergio Serra https://orcid.org/0000-0002-0792-8157
- Xuanqi Li https://orcid.org/0000-0003-1498-6205