

# DataCrate: a method of packaging, distributing, displaying and archiving Research Objects

Peter Sefton, University of Technology Sydney, peter.sefton@uts.edu.au

Gerard Devine, Western Sydney University, g.devine@wsu.edu.au

Christian Evenhuis, University of Technology Sydney, christian.evenhuis@uts.edu.au

Michael Lynch, University of Technology Sydney, michael.lynch@uts.edu.au

Sharyn Wise, University of Technology Sydney, sharyn.wise@uts.edu.au

Michael Lake, University of Technology Sydney, mike.lake@uts.edu.au

Duncan Loxton, University of Technology Sydney, duncan.loxton@uts.edu.au

**Abstract**—DataCrate is a specification for packaging research data for distribution and reuse. DataCrate packages may be distributed as archives with or without compression, and hosted on web sites, as they contain an index.html page that summarizes the contents of the package for human readers. For machines, there is a JSON-LD file containing metadata that will aid in the re-use of the data including licensing, publication dates, parties involved in creating the data. Where possible the terms from schema.org are used for metadata, with other ontologies used where needed.

DataCrate optionally uses the BagIt data packaging standard which is widely used in libraries and archives and increasingly in research data management, including in recent discussions about data packaging at the Research Data Alliance (RDA).

**Index Terms**—Research Objects; Data Packaging; Metadata

## I. INTRODUCTION

In characterizing the term *Research Object* the call for proposals for Research Object 2018 uses the phrase “multi-part research outcomes with their context”. The DataCrate specification<sup>1</sup> is a research data packaging and dissemination specification designed to capture exactly that; outcomes (also inputs) and context.

DataCrate specifies how to gather together data in such a way that it can (a) be packaged via zip, tar, a disc image, a multi-part package or (b) be hosted on a web server or file share for inspection by potential users and/or used directly on High Performance Computing systems or otherwise accessed and analyzed.

DataCrates can contain any kind of data, and the contextual information may include, but is not limited to, data about the people, software and equipment used in the research as well as supporting documents such as publications, funding agreements or README files.

## II. METHODOLOGY

The DataCrate specification grew out of two generations of previous data packaging work. The first implementation was in the HIEv system at Western Sydney University [1], a data capture system for environmental science data that captures data files produced by sensor networks and allows manual or API-based upload of other files. Using a web interface researchers can select files to export, for example to support an research article.

The requirements / principles for the HIEv system were presented at eResearch Australasia.

1. The packaging format should not be data-format-sensitive.
2. The packaging format should not be research domain specific.
3. The packaging format should not be technology or platform specific.
4. The data package should contain as much contextual information as possible.
5. Metadata should be easily human and machine-readable.
6. The package format should contain self-checking and verification features.
7. The metadata format should be compatible with the semantic web by using URIs as names for things including metadata terms\*.
8. Requirement 7 implies using Linked-Data [2], but the project should not attempt to define and manage its own ontologies, for reasons of sustainability\*.
9. A data package should be able to be displayed on the web - implying that the human readable metadata in 5 should be in HTML\*.

\*The last three requirements or principles were not explicit in the presentation but were discussed during the development, and proved increasingly important in the development of the current DataCrate specification.

The data packaging in HIEv used the Bagit [3] packaging spec to cover requirement 6 - BagIt also doesn't get in the

<sup>1</sup><https://github.com/UTS-eResearch/DataCrate/>

way of any of the other requirements.

The main innovation in HIEv’s packaging was to add an HTML file that covered requirements 4 (as much context as possible) & 5 (human and machine readable metadata). To do this, HIEv produces a summary of the context, with information about the facilities used – their name, nature and location – and technical details about the payload files, thus satisfying requirement 5. Using RDFa [4] to embed metadata in the HTML file gave both human (HTML) and machine (RDFa) views of the data.

Cr8it [5] was another early implementation that used the same basic idea for data packages in the OwnCloud file sharing application.

The first two DataCrate proto-implementations had no guidelines for what metadata to use beyond what was hard-wired into each code-base, so there was no hope of easy interoperability or safe extensibility, and there were no repositories into which data could be published, but we had good feedback about the concept from the eResearch community and from the very limited number of researchers exposed to the systems, so in 2016 when UTS began work on a new Research Data Management service [6], we decided to properly specify a data packaging format that met the above requirements and the DataCrate standard was born.

A team based at UTS, with some external collaborators started a process to work out (a) was there an existing standard which had emerged since the HIEv work that met the requirements? (b) If not, which RDF vocabularies should we use? and (c) the mechanics of organizing the files in the packages. At this point our specification had matured and we were looking to find or build a data packaging format which had the following *DataCrate Requirements*:

1. Checksums per-file and the ability to include linked resources (features of BagIt)
2. A linked-data metadata encoding of requirement 2 in JSON-LD format using well-documented ontologies / vocabularies with coverage for:
  - Discovery and citation metadata; **who** created it, **what** is it, **where** did the work take place, **what** in the world is it *about* for both the package/dataset and the file level.
  - Technical metadata; what size is each file, what format is the file in, who or what created the file.
  - Provenance for files; how were they created, with what equipment and software.
3. A convention for including an HTML file which describes the dataset, and potentially all of its files with a human-readable view of requirement 2.

These could all be accomplished by an update of the HIEv data package but it was important to make sure we were not re-inventing something that had been done elsewhere.

1) *Existing standards*: We were not able to find any general-purpose packaging specification with anything like the HTML+RDFa index that HIEv data packages

have, allowing for human and machine readable metadata. Using BagIt plus extra files worked well in our initial implementations so unless a better alternative surfaced – the decisions were around formalizing metadata standards.

BagIt, which had been used in HIEv and Cr8it, is an obvious standard on which to base a research data packaging format - it is widely used in the research data community, there is cross-platform (requirement 3) tooling available and it covers the integrity aspects of packaging data.

2) *Alternatives considered*: Two data packaging options were identified as mature enough to be evaluated: Frictionless Data Packages and Research Objects.

Frictionless Data Packages [7], which uses a simple JSON format as a manifest, has roughly equivalent packaging features to BagIt, having checksum features built in. In their favor, Frictionless Data Packages have the ability to describe the headers in tabular data files. However, they do not meet the requirement 7 of having linked-data metadata, so while the JSON metadata is technically machine readable, in that is simple to parse, it is not easy to relate to the semantic web as it does not use linked-data standards. Further the terms are defined locally to the specification, without URIs. It is also unclear how to extend the specification in a standardized way, contrasting with linked-data approaches which *automatically* allow extension by the use of URIs.

As an example, the specification<sup>2</sup> does not give a single way to describe temporal coverage:

Adherence to the specification does not imply that additional, non-specified properties cannot be used: a descriptor MAY include any number of properties in addition to those described as required and optional properties. For example, if you were storing time series data and wanted to list the temporal coverage of the data in the Data Package you could add a property temporal (cf Dublin Core):

```
“temporal”: { “name”: “19th Century”, “start”:
“1800-01-01”, “end”: “1899-12-31” }
```

This flexibility enables specific communities to extend Data Packages as appropriate for the data they manage. As an example, the Tabular Data Package specification extends Data Package to the case where all the data is tabular and stored in CSV. <https://github.com/frictionlessdata/specs/blob/0860ecd6bbb7685425e6493165c9b1a1c91eb16b/specs/data-package.md>

This *laissez faire* extension mechanism in Frictionless Data Packages has appeal, but it is likely to result in a proliferation of highly divergent non-standardized metadata, jeopardizing interoperability. DataCrate aims

<sup>2</sup><https://frictionlessdata.io/specs/data-package/>

to encourage common behaviors, to facilitate metadata interchange by using JSON-LD and by specifying widely used metadata standards. So, using the temporal coverage example, DataCrate uses schema.org’s temporalCoverage property. schema.org’s selection certainly raised some eyebrows in the library world and it is far from perfect, but its adoption by major search engines led us to believe that it would achieve a ubiquity that would stand DataCrate in good stead. Indeed, this choice means that as services such as Google’s Dataset search roll out, DataCrates will be indexed, whereas the semantics of FrictionlessData packaged are less likely to be indexed in detail. See the details provided by Google<sup>3</sup>.

```
{
  "@id": "https://doi.org/10.5281/zenodo.1009240",
  "@type": "Dataset",
  <...>
  "name": "Sample dataset for DataCrate v1.0",
  "publisher": {
    "@id": "http://uts.edu.au"
  },
  "temporalCoverage": "2017"
}
```

The ability to resolve URIs is an important one if the CATALOG.html file is to be useful to humans. Many URIs in scientific and other ontologies do not resolve to a webpage but to an ontology file, which is unhelpful for people trying to understand what metadata terms *mean*, so in DataCrate we introduced a mechanism for mapping non-human-friendly URIs to useful ones.

For example. This URI from the [BIBO] ontology <“http://purl.org/ontology/bibo/interviewee”> resolves to an ontology file which downloads as `bibo.php`. A user would then have to locate the definition, embedded in an `rdfs:comment` element in that file:

```
<rdfs:comment xml:lang="en">
An agent that is interviewed
by another agent.
</rdfs:comment>
```

By contrast, `http://neologism.ecs.soton.ac.uk/bibo.html#interviewee` is human-readable; the DataCrate solution is to map the “official”, machine readable URI to the more useful one, using `schema:sameAs` and to link the term “interviewee” in the *DataCrate Website* to the useful URL, for the use of humans. The HTML in the *DataCrate Website* is self-documenting:

**interviewee?**      Anita de Waard

Figure 1. Screenshot showing how the term `temporalCoverage` is linked - the ? link resolves to a page which documents the `interviewee`.

<sup>3</sup><https://developers.google.com/search/docs/data-types/dataset>

The other main alternative evaluated was the Research Object Bundle specification [8].

Rather than BagIt, the original version of Research Object Bundle used the Zip-based Universal Container Format - a format for which the documentation now seems to be unavailable from Adobe. It does not have integrity features such as checksums but there is a version of Research Object which uses BagIt<sup>4</sup>.

RO BagIt *does* use Linked-Data and for that reason was given careful consideration as a base-format for DataCrate. However, there were some implementation details that we thought would make it harder for tool-makers (including the core team at UTS); the use of “aggregations” and “annotations” introduces two extra layers of abstraction for describing resources.

For example, using the sample from the `bagit-ro` tool, there is a section in the manifest that lists aggregated files:

```
"aggregates": [
  { "uri": "../data/numbers.csv",
    "mediatype": "text/csv"
  }, ..
]
```

And a separate place to describe *annotations* on those files:

```
"annotations": [
  {
    "about": "../data/numbers.csv",
    "content": "annotations/numbers.jsonld",
    "createdBy": {
      "name": "Stian Soiland-Reyes",
      "orcid": "http://orcid.org/0000-0001-9842-9718"
    }
  }
]
```

With the actual description of the `numbers.csv` file residing in `annotations/numbers.jsonld`.

```
{ "@context": { "@vocab": "http://purl.org/dc/terms/",
  "dcmi": "http://purl.org/dc/dcmitype/Dataset"},
  "@id": "../data/numbers.csv",
  "@type": "dcmi:Dataset",
  "title": "CSV files of beverage consumption",
  "description": "A CSV file <...>"
}
```

In our judgement, the level of indirection and number of files involved in the Research Object approach were not suitable for DataCrate as the implementation cost for tool makers would be too high. In making this choice we forfeited the benefits of being able to make assertions about the provenance of annotations as distinct resources, and the more intellectually satisfying abstractions about aggregations offered by ORE.

<sup>4</sup><https://github.com/ResearchObject/bagit-ro>

### III. IMPLEMENTATION OF DATACRATE

We settled on an approach which used:

1. A single CATALOG.json file, containing JSON-LD which describes the folder/file hierarchy of the data crate and associated contextually relevant entities, such as people, all in one place.
2. A CATALOG.html file which is an entry point to a *DataCrate Website* with a human-readable summary of the catalog file, one item per page.
3. Optionally, a DataCite.xml file containing a data citation (a text version of which is prominent in the HTML file if it exists).

Uses for DataCrates planned at UTS include: - Making DataCrates available for download from a public website, both packaged as Zip files and expanded so that users can peruse the *DataCrate Website* via the CATALOG.html file and access individual files. - Using the DataCrate with additional metadata for archiving and preserving data (in a project to begin in 2019). - Using the DataCrate format to allow exchange of data between systems, for example sending data from a repository in a university facility such as the Omero microscopy repository [9] to a git project system like GitLab. - Automatically detecting metadata in DataCrates which are uploaded to our research-integrity driven Research Data Management system [6].

The initial version of DataCrate (v0.1) was developed in 2017. V0.1 persisted with HTML+RDFa for human and machine readability but this was cumbersome to generate and was removed at [the suggestion of Eoghan Ó Carragáin][(<https://github.com/UTS-eResearch/DataCrate/issues/14>)]<sup>5</sup> in favor of an approach where the human-centred HTML page is generated from a machine-readable JSON-LD file rather than the other way around.

Prior to selecting schema.org, we evaluated at a variety of standards, including Dublin Core [10] which is very limited in coverage and DCAT [11] which is more complete for describing datasets at the top level, but silent on the issue of describing files or other contextual entities and relationships between them. We discovered that schema.org has the widest range of terms needed to describe “who, what, where” metadata for datasets. Over the course of more than a year we have worked through the process of creating DataCrates for data from a variety of disciplines and found that schema.org has the coverage to meet the requirement.

Some examples are:

- Data relating to the IDRC funded project (described in <https://doi.org/11.0897/rio.2.e8880>) to examine data management policies and implementation for

development funders<sup>6</sup>. This example shows the use of the schema.org `sameAs` property to provide a human readable gloss for the key ‘interviewee’

- Some Matlab code<sup>[^Matlab code]</sup> that supports a research article. This is a good illustration of how common names like Lu, J can be disambiguated using [ORCID] identifiers.
- Victoria Arch<sup>7</sup> contains data from a 3d survey of a cave, conducted using a Lidar scanner mounted on a drone. This example contains examples of data-provenance, via the use of schema.org `CreateAction` properties that show that files result from observation-actions on the `object` of study (the cave).
- A sample data set<sup>8</sup>.
- Some clinical trial data<sup>9</sup> showing how researcher affiliations can be modeled using linked data. See an example of a Person with four affiliations.
- For describing datasets such as exported content from digital object repository systems DataCrate uses the Portland Common Data Model [12], (PCDM), which is a simple ontology for describing nested Collections of Objects, with Objects having Files. See an example of a complete social-history repository which has been exported to DataCrate using PCDM to model its structure<sup>10</sup>.

### IV. TOOLS

There are a number of tools for DataCrate in development.

At the University of Technology Sydney [6], the Provisioner is an open framework for integrating good research data management practices into everyday research workflows. It provides a user-facing research data management planning tool which allows researchers to describe and publish datasets and create and share workspaces in different research apps such as lab notebooks, code repositories (where data is included by-reference), survey tools and collection management tools. DataCrates are used as an interchange format to move data between the different research apps, and as an ingest, archive and publication format. Lightweight adaptors coded against each research app’s native API allow export and import of DataCrates, which are then used to move data from one app to another, while recording a provenance history in the DataCrates’ metadata. Examples of DataCrates moving through the research lifecycle will be provided.

HIEV DataCrate - At the Hawkesbury Institute for the Environment at Western Sydney University, HIEV harvests a wide range of environmental data (and associated file level metadata) from both automated sensor networks and

<sup>6</sup>[https://data.research.uts.edu.au/examples/v1.0/Data\\_Package-IDRC\\_Opportunities\\_and\\_Challenges\\_Open\\_Research\\_Strategies](https://data.research.uts.edu.au/examples/v1.0/Data_Package-IDRC_Opportunities_and_Challenges_Open_Research_Strategies) [<sup>^</sup>Matlab code]: <https://data.research.uts.edu.au/examples/v1.0/GTM/>

<sup>7</sup>[https://data.research.uts.edu.au/examples/v1.0/Victoria\\_Arch\\_pub/](https://data.research.uts.edu.au/examples/v1.0/Victoria_Arch_pub/)

<sup>8</sup><https://data.research.uts.edu.au/examples/v1.0/sample/>

<sup>9</sup><https://data.research.uts.edu.au/examples/v1.0/timluckett/>

<sup>10</sup>[https://data.research.uts.edu.au/examples/v1.0/farms\\_to\\_freeways/](https://data.research.uts.edu.au/examples/v1.0/farms_to_freeways/)

<sup>5</sup><https://github.com/UTS-eResearch/DataCrate/issues/14>

analysed datasets generated by researchers. Leveraging built-in APIs within the HIEv a new packaging function has been developed<sup>11</sup>, allowing for selected datasets to be identified and packaged in the DataCrate standard, complete with metadata automatically exported from the HIEv metadata holdings into the JSON-LD format. Going forward this will allow datasets within HIEv to be published regularly and in an automated fashion, in a format that will increase their potential for reuse.

MIF DataCrate - The Microbial Imaging Facility (MIF) is part of the ithree institute and School of Life Sciences at University of Technology Sydney. It supports 100 active researchers by providing training, access and support for ten microscopes. There is broad range of equipment, which covers four major brands, and ranges from basic optical microscopes to cutting edge super-resolution. Handling the images and their associated metadata from the array of instruments is made possible through the effort of the Open Microscopy Environment. Specifically, the MIF extensively uses:

- Bioformats software [13] which enables a wide range of proprietary image formats to be read.
- OME-XML [14], a schema that can describes how microscope are setup.
- OMERO [15], a client-server software for storing, sharing, viewing, organizing and analyzing microscopy images.

While the OME comprehensively covers the metadata associated with images, it is necessary to extend for including facility-specific and broader experimental metadata. Several tools have developed to integrate OMERO with DataCrate, including scripts for:

- adding external metadata and to manage internal metadata in OMERO<sup>12</sup>
- exporting images and metadata from OMERO<sup>13</sup>
- integrating imaging data with other experimental data with a DataCrate
- identity management across systems

Calcytejs<sup>14</sup> is a command line tool for packaging data into DataCrate being developed at the University of Technology Sydney which allows researchers to describe any data set via the use of spreadsheets which the tool auto-creates in a directory tree.

Omeka DataCrate Tools<sup>15</sup> is Python tool to export data from Omeka Classic repositories into the DataCrate format and to import them into Omeka S.

<sup>11</sup>[https://github.com/gdevine/hiev\\_DataCrate](https://github.com/gdevine/hiev_DataCrate)

<sup>12</sup><https://github.com/evenhuis/omero-user-scripts>

<sup>13</sup><https://code.research.uts.edu.au/MIF/Workflows/omero-DataCrate>

<sup>14</sup><https://code.research.uts.edu.au/eresearch/CalcyteJS>

<sup>15</sup><https://github.com/UTS-eResearch/omeka-DataCrate-tools>

DSpace to DataCrate<sup>16</sup> is an early-stage Nodejs tool for extracting repository data from DSpace repositories into DataCrate format.

#### A. Extending DataCrate for use in a specific domain - a case study in microscopy data

There are difficulties in using discipline-specific linked data. To illustrate this we will look at a case study from the Microbial Imaging Facility at UTS. The goal is to package microscope-captured images with detailed provenance information, such as the microscope model, lens, light source, and filters used to make an image, as the most important requirement is that a lab technician can re-set a microscope to the same configuration as was used to capture an image-set.

There is a well established standard for recording this information, using The Open Microscopy Environment (OME) Data Model in XML [16], but to find DataCrate-friendly URIS for classes and properties we tried searching in Bioportal [17]. It became apparent that trying to pick URIs would involve choosing between and using terms from several ontologies to stitch-together a JSON `@context`; this approach would have involved a lot of detailed investigation and maintenance.

Instead, the solution to this we have adopted in DataCrate is to create a web page for each element and attribute in the OME schema and to use those in combination with `schema.org`.

Firstly, make sure that the most important information is captured using the `schema.org` `CreateAction` class. That is, the act of creation is recorded with links the the instruments used.

Each piece of equipment has its own URI, resolving to a page, and is listed in the *DataCrate-flattened JSON-LD* with as much detail as possible. Each scientific instrument will have at least the `@type` `IndividualProduct` but may also have additional types and properties from the OME schema, via the pages we created, and could also include URIs from other ontologies if that will aid in understanding for users, or discovery. An example is available in the DataCrate sample<sup>17</sup> dataset.

## V. ISSUES WITH SOFTWARE TOOLING

There is a distinct lack of simple to use JSON-LD tools for programers. While there are libraries for JavaScript and Python (the languages the authors use most frequently) to do high level operations such as normalizing, framing or flattening JSON documents, we have not found libraries with utility functions such the ability to resolve context keys, which is a non-trivial programming task, or to traverse

<sup>16</sup><https://github.com/UTS-eResearch/DataCrate-dspace-tools>

<sup>17</sup><https://data.research.uts.edu.au/examples/v1.0/sample/>

the JSON `@graph`. Thus, the DataCrate specification mandates that JSON-LD must be organized in a particular, predictable way (flattened) and the the ??? is a simple mapping of key-to-URI with no indirection.

## VI. CONCLUSION

DataCrate has just reached version one as of the fourth quarter of 2018, and has not yet been widely used, so it is impossible to evaluate its success. Anecdotal feedback from conferences and researchers has been positive, but we will need to evaluate the usability and utility of the system when significant numbers of packages have been distributed and used.

In conclusion, we offer some proposals for further discussion and possible action.

- An effort is needed to produce libraries to make consuming JSON-LD easy for developers, libraries for (at least) Python, Javascript and R that can load arbitrary JSON-LD, resolve its context and vocabularies, and provide an index and graph traversal.
- More work should be undertaken to align DataCrate with preservation workflows.
- The DataCrate specification should be extended to allow for the description of the contents of files, such as column headers in tabular data formats.
- As specification developers we note that working with schema.org and JSON-LD has been a straightforward process (lack of library support notwithstanding), it is easy to find and use schema.org terms and DataCrate approach of using Flattened JSON-ID, with a simplified context has been productive for development. Our recommendation to research communities wanting to embrace linked data, and the benefits of the human-readable metadata in data crates, and where there is no established, usable ontology would be to explore the idea of creating a schema.org external extension for their discipline or technical specialization.

## REFERENCES

- [1] P. Sefton and P. Bugeia, “Introducing next year’s model, the data-crate; applied standards for data-set packaging,” in *eResearch Australasia 2013*, 2013[Online]. Available: [http://eresearchau.files.wordpress.com/2013/08/eresau2013\\_submission\\_57.pdf](http://eresearchau.files.wordpress.com/2013/08/eresau2013_submission_57.pdf).
- [2] T. Berners-Lee, *Linked data, 2006*. [Online]. Available: <http://www.w3.org/DesignIssues/LinkedData.html>.
- [3] J. Kunze, A. Boyko, B. Vargas, L. Madden, and J. Littman, “The BagIt File Packaging Format (V0.97).” [Online]. Available: <http://tools.ietf.org/html/draft-kunze-bagit-06>. [Accessed: 01-Mar-2013].
- [4] “RDFa,” *Wikipedia, the free encyclopedia*, Feb. 2014[Online]. Available: <http://en.wikipedia.org/w/index.php?title=RDFa//&oldid=592600280>. [Accessed: 26-Feb-2014].
- [5] P. Sefton, P. Bugeia, and V. Picasso, “Pick, Package and Publish research data: Cr8it and Of The Web,” in *eResearch Australasia 2014*, 2014[Online]. Available: [http://eresearchau.files.wordpress.com/2014/07/eresau2014\\_submission\\_30.pdf](http://eresearchau.files.wordpress.com/2014/07/eresau2014_submission_30.pdf).
- [6] L. Wheeler, S. Wise, and P. Sefton, “End-to-End Research Data Management for the Responsible Conduct of Research at the University of Technology Sydney,” 2018[Online]. Available: [https://eresearch.uts.edu.au/2018/07/04/APRI\\_2018\\_provisioner.htm](https://eresearch.uts.edu.au/2018/07/04/APRI_2018_provisioner.htm). [Accessed: 10-Jul-2018].
- [7] “The Frictionless Data Field Guide.” [Online]. Available: <https://frictionlessdata.io/specs/data-package/>. [Accessed: 26-Sep-2018].
- [8] S. Soiland-Reyes, M. Gamble, and R. Haines, “Research object bundle 1.0,” *Specification, researchobject. org*, 2014.
- [9] “OMERO: Flexible, model-driven data management for experimental biology | Nature Methods.” [Online]. Available: <https://www.nature.com/articles/nmeth.1896>. [Accessed: 13-Jul-2018].
- [10] J. Kunze and T. Baker, “The Dublin Core Metadata Element Set,” 2007[Online]. Available: <http://www.rfc-editor.org/info/rfc5013>. [Accessed: 12-Jul-2018].
- [11] F. Maali, J. Erickson, and P. Archer, “Data catalog vocabulary (DCAT),” *W3C Recommendation*, vol. 16, 2014[Online]. Available: <https://www.w3.org/TR/2014/REC-vocab-dcat-20140116/>.
- [12] “Portland Common Data Model.” [Online]. Available: <https://pcdm.org/2016/04/18/models>. [Accessed: 12-Jul-2018].
- [13] M. Linkert, C. T. Rueden, C. Allan, J.-M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. MacDonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. W. Eliceiri, and J. R. Swedlow, “Metadata matters: Access to image data in the real world,” *The Journal of Cell Biology*, vol. 189, no. 5, pp. 777–782, 2010[Online]. Available: <http://jcb.rupress.org/content/189/5/777>.
- [14] I. G. Goldberg, C. Allan, J.-M. Burel, D. Creager, A. Falconi, H. Hochheiser, J. Johnston, J. Mellen, P. K. Sorger, and J. R. Swedlow, “The open microscopy environment (ome) data model and xml file: Open tools for informatics and quantitative analysis in biological imaging,” *Genome Biology*, vol. 6, no. 5, p. R47, May 2005[Online]. Available: <https://doi.org/10.1186/gb-2005-6-5-r47>.
- [15] C. Allan, J.-M. Burel, J. Moore, C. Blackburn, M. Linkert, S. Loynton, D. MacDonald, W. J. Moore, C. Neves, A. Patterson, and others, “OMERO: Flexible, model-driven data management for experimental biology,” *Nature methods*, vol. 9, no. 3, p. 245, 2012.
- [16] I. G. Goldberg, C. Allan, J.-M. Burel, D. Creager, A. Falconi, H. Hochheiser, J. Johnston, J. Mellen, P. K.

Sorger, and J. R. Swedlow, “The Open Microscopy Environment (OME) Data Model and XML file: Open tools for informatics and quantitative analysis in biological imaging,” *Genome Biology*, vol. 6, no. 5, p. R47, May 2005[Online]. Available: <https://doi.org/10.1186/gb-2005-6-5-r47>. [Accessed: 26-Sep-2018].

[17] P. L. Whetzel, N. F. Noy, N. H. Shah, P. R. Alexander, C. Nyulas, T. Tudorache, and M. A. Musen, “BioPortal: Enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications,” *Nucleic Acids Research*, vol. 39, no. suppl\_2, pp. W541–W545, Jul. 2011[Online]. Available: [https://academic.oup.com/nar/article/39/suppl\\_2/W541/2507188](https://academic.oup.com/nar/article/39/suppl_2/W541/2507188). [Accessed: 26-Sep-2018].