

Databases and ontologies

libOmexMeta: enabling semantic annotation of models to support FAIR principles

Ciaran Welsh ¹, David P. Nickerson², Anand Rampadarath², Maxwell L. Neal³, Herbert M. Sauro¹ and John H. Gennari ^{4,*}

¹Department of Bioengineering, University of Washington, 98195 Seattle, WA, USA, ²Auckland Bioengineering Institute, University of Auckland, 1142 Auckland, New Zealand, ³Center for Global Infectious Disease Research, Seattle Children's Research Institute, 98101 Seattle, WA, USA and ⁴Department of Biomedical Informatics and Medical Education, University of Washington, 98019 Seattle, WA, USA

*To whom correspondence should be addressed.

Associate Editor: Peter Robinson

Received on January 21, 2021; revised on May 12, 2021; editorial decision on June 11, 2021; accepted on June 14, 2021

Abstract

Summary: As the number and complexity of biosimulation models grows, so do demands for tools that can help users better understand models and make those models more findable, shareable and reproducible. Consistent model *annotation* is a step toward these goals. Both models and tools are written in a variety of different languages; thus, the community has recognized the need for standard, language-independent methods for annotation. Based on the Computational Modeling in Biology Network community consensus, we introduce an open-source, cross-platform software library for semantic annotation of models.

Availability and implementation: libOmexMeta is freely available at <https://github.com/sys-bio/libOmexMeta> under the Apache License 2.0. A live demonstration is at github.com/sys-bio/pyomexmeta-binder-notebook.

Contact: gennari@uw.edu

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Biosimulation models are used throughout the life sciences to test hypotheses about biological systems and explore the effects of perturbations on those systems. These models are written in a variety of languages, and they can simulate a wide variety of biological processes across physical scales. As models have proliferated, model-archiving initiatives such as BioModels (Glont *et al.*, 2017) and the Physiome Model Repository (Yu *et al.*, 2011) have been established to make models public and more accessible. Although these repositories are a commendable first step, searching across repositories, or integrating models originating in different repositories remain cumbersome, sometimes nearly impossible tasks. We argue that the worldwide collection of publicly available models should meet the criteria put forth as the FAIR principles: research should be *Findable, Accessible, Interoperable* and *Reusable* (Wilkinson *et al.*, 2016). Thus, to make models more findable and reusable, they must be annotated with common semantics against identifiers from standards and bio-ontologies such as the Gene Ontology, UniProt and ChEBI.

The biosimulation research community includes a consortium of researchers focused on developing standards for modeling, known as the Computational Modeling in Biology Network (COMBINE). This group has recently developed a consensus around harmonizing

approaches to annotation in a way that is language-independent (Neal *et al.*, 2019). Although this consensus led to the development of a specification of how to annotate, this is not enough to expect annotation to happen ‘in the wild’. To make any artifact shareable and reusable requires more time and effort than it does to build the artifact for a single use by a single lab. To reduce this cost, we need tools that make it as easy as possible to annotate models.

Along with the proliferation of biosimulation models, there has been an accompanying growth in the number of tools available to work with those models. These tools provide support for the entire lifecycle of biosimulation modeling, from model creation, to execution of the model, to parameter tuning and to model refinement and validation. Many of these tools provide some support for model annotation, and recently, some include the ability to read and write standard archive files that conform to COMBINE's Open Modeling and Exchange (OMEX) format (Bergmann *et al.*, 2014). However, these tools do not conform to the recent consensus on how to best annotate models (Neal *et al.*, 2019). Consequently, current annotations on models use inconsistent styles and are language-specific.

Our solution is not to define yet another tool for working with biosimulation models, but instead, the development of a software library that can easily be incorporated into all of these tools. The library is designed for tool developers, and allows them to

consistently create, edit and work with annotations on biosimulation models. By ‘consistently’, we mean consistent with the community-consensus OMEX Metadata Specification, which describes in detail our recommendations for annotations. The [Supplementary Material](#) includes v1.1 of this specification for the metadata that describes a biosimulation model. In this article, we describe our implementation of this specification via the libOmexMeta library and its Python frontend package, pyomexmeta.

2 The libOmexMeta library

libOmexMeta is a C++ library available from <https://github.com/sys-bio/libOmexMeta/> under an Apache 2.0 license. A goal of our implementation was portability, and thus we choose C++ as this allows us to port libOmexMeta to other languages via a shared library. We demonstrate this functionality via a Python package, pyomexmeta, which uses the libOmexMeta shared library to make libOmexMeta features available from a Python environment. Pyomexmeta is available from pypi.org using ‘pip install pyomexmeta’, and we provide examples and a live demonstration of pyomexmeta at github.com/sys-bio/pyomexmeta-binder-notebook.

libOmexMeta depends on the Redland RDF packages: raptor2, rasqal and librdf (<http://librdf.org/>). libOmexMeta enables users and tools to read and write annotations to and from a variety of syntaxes. Users can read RDF/XML either directly on its own or when embedded within another XML document. Other supported syntaxes include turtle, ntriples, nquads, html and json. Features of our RDF-based implementation are that resulting annotations can be queried using SPARQL and can be visualized from the Python frontend using graphviz. In addition, users can optionally store annotations in a sqlite3 database.

3 Annotating models

The library provides the ability to create a variety of standard annotations. These can include model-level annotations, such as specifying the author(s) of a model, or more fine-grained annotations such as specifying the participants in a particular biochemical reaction in a model. For example, one could annotate a reaction as having sources, sinks and mediators (i.e. reactants, products and enzymes), as well as a characterization of the entire reaction such as via a Gene Ontology term.

As described in the OMEX Metadata Specification, each annotation is a connection between some bio-ontology resources (e.g. the Gene Ontology or ChEBI) and a metaID that points to a specific codeword with the accompanying source code. Thus, using Python and pyomexmeta, one can create a reaction annotation as follows:

```
with editor.new_physical_process() as physical_process:
    physical_process\
        .about("Omex_R001", eUriType.MODEL_URI)\
            .is_version_of("GO:0004022",
                eUriType.IDENTIFIERS_URI)\
        .add_source("Omex_S001", eUriType.MODEL_URI)\
        .add_sink("Omex_S002", eUriType.MODEL_URI)\
        .add_mediator("Omex_S003", eUriType.MODEL_URI)
```

Here, each of the Omex_Sxxx are pointers to physical entities (species) in the corresponding model source code (whether that be SBML, CellML or other) that encode the participants in the reaction, and the reaction as a whole is indicated by ‘Omex_R001’. The ‘is_version_of’ annotation indicates that this reaction is a version of the GO process of alcohol dehydrogenase (ID #0004022). The actual reaction can have other mediators, and these can be easily added with additional ‘.add_mediator’ statements.

A portion of the resulting annotations (shown in RDF turtle format and using the model name of ‘test.xml’) can be seen below:

```
<http://omex-library.org/test.omex/test.xml#Omex_R001>
semsim:hasMediatorParticipant local:MediatorParticipant001;
```

```
semsim:hasSourceParticipant local:SourceParticipant002;
semsim:hasSinkParticipant local:SinkParticipant003;
qbio:isVersionOf <http://identifiers.org/go/GO:0004022>.
```

As further described in the [Supplementary Material](#), the ‘local:’ pointers are needed to store information about that particular participant, as a given protein may participate in more than one reaction. The [Supplementary Material](#) also provides many other examples of annotation using pyomexmeta. For example, we distinguish among model-level annotations (such as author), singular annotations for components (such as ChEBI identifiers for small molecules) and composite annotations that require more than one resource (such as for reactions). For composite annotations, we provide examples of entities with physical properties (such as volumes), properties of processes (such as their participants) and properties of forces (such as electrical potentials).

libOmexMeta also includes the ability to read existing annotations in SBML files. Thus, for SBML models that include annotations about species and reactions, libOmexMeta can transform this information into the appropriate RDF file of language-independent annotations. As we describe next, these annotation files can then be stored as part of an OMEX archive for improved search and model retrieval.

4 Summary and discussion

As an initial use of our library, and in collaboration with the BioModels curators, we have begun to apply libOmexMeta comprehensively to all of the curated models in the BioModels.org collection (1000 models). These updated OMEX archives will support much finer-grained searching than is currently possible. For example, ‘retrieve all models that include a reaction where enzyme phosphofructokinase is a mediator, and ADP is a product’. This sort of search cannot be carried out by the current library, yet is straightforward to implement in SPARQL (the query language for RDF), given a graph representation of the RDF from these libOmexMeta annotation files from BioModels. A key feature of our approach is that this graph and the resulting search capability are independent of the modeling language. As an important test case, we have annotated a number of CellML models and have demonstrated that these can also be searched along with models written in SBML.

In addition to users in the BioModels group, we have been working closely with developers of model development tools such as OpenCOR ([Garny and Hunter, 2015](#)), COPASI ([Hoops et al., 2006](#)) and Tellurium ([Choi et al., 2018](#)). Developers from each of these systems attended a libOmexMeta tutorial (at the HARMONY meeting, 2021) to better understand how to incorporate libOmexMeta. Importantly, the COMBINE community has ratified this approach to annotation; thus, tool developers will need to support model annotation via libOmexMeta.

To make models more *findable*, they should have annotations that refer to terms from standard bio-ontologies. To allow searching to be independent of modeling language, these annotations should be stored separately from the source code. As described in [Neal et al. \(2019\)](#), the best way to support language-independent search is to use a separate file, with annotations stored in a consistent format (e.g. RDF). Otherwise, search tools would have to parse and understand multiple languages.

To make models more *accessible*, annotations should help researchers better understand the biological semantics of models. These semantics include both model-level annotations (e.g. about the high-level biological process being modeled) and variable-level semantics (e.g. about a specific protein in a specific reaction). Additionally, annotations can provide non-biological information about the model, such as publication, author, etc. In general, model-level annotations are the focus of the proposed ModeXchange initiative ([Hermjakob and Malik-Sheriff, 2021](#)). In a complementary fashion, the OMEX metadata specification focuses on variable-level annotations, and libOmexMeta is designed to support both types of annotation. To make models more *interoperable*, we advocate using annotations against standard ontologies. Finally, models can be *reusable* only if the semantics of model terms and variables are well-defined and consistent.

libOmexMeta need not be the only implementation of the OMEX metadata standard; in fact, we are also developing a Java library for tool developers that may prefer to work in that environment. Regardless of implementation, once a collection of models are annotated and made available as OMEX archives, then standard RDF query tools (SPARQL) can easily search over the entire graph of model annotations. Thus, the OMEX metadata specification provides a roadmap for the application of FAIR principles to modeling, and the libOmexMeta library provides an implementation of these ideas that is easy for tool-builders to incorporate.

This work has been funded by the National Institutes of Health [grant P41 GM109824].

Conflict of Interest: none declared.

References

- Bergmann, F.T. et al. (2014) COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics*, 15, 369.
- Choi, K. et al. (2018) Tellurium: an extensible python-based modeling environment for systems and synthetic biology. *BioSystems*, 171, 74–79.
- Garny, A. and Hunter, P.J. (2015) OpenCOR: a modular and interoperable approach to computational biology. *Front. Physiol.*, 6, 26.
- Glont, M. et al. (2017) BioModels: expanding horizons to include more modeling approaches and formats. *Nucleic Acids Res.*, 46, D1248–D1253.
- Hermjakob, H. and Malik-Sheriff, R.S. (2021) ModeleXchange—status update and data invitation. In *HARMONY Meeting*.
- Hoops, S. et al. (2006) COPASI—a COmplex PATHway Simulator. *Bioinformatics*, 22, 3067–3074.
- Neal, M.L. et al. (2019) Harmonizing semantic annotations for computational models in biology. *Brief. Bioinformatics*, 20, 540–550.
- Wilkinson, M.D. et al. (2016) Comment: the FAIR guiding principles for scientific data management and stewardship. *Sci. Data*, 3, 9.
- Yu, T. et al. (2011) The Physiome Model Repository 2. *Bioinformatics (Oxford, England)*, 27, 743–744.