SED-ML Validator: tool for debugging simulation experiments

Bilal Shaikh [1,*], Andrew Philip Freiburger [2,*], Matthias König [3], Frank T. Bergmann [4,5], David P. Nickerson [6], Herbert M. Sauro [7], Michael L. Blinov [8], Lucian P. Smith [7], Ion I. Moraru [8] and Jonathan R. Karr [1,†]

[1]Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY 10029, USA, [2]Department of Civil Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada, [3]Department of Theoretical Biology, Humboldt University, 10115 Berlin, Germany, [4]BioQUANT/COS, Heidelberg University, 69120 Heidelberg, Germany, [5]Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena 91125, CA, USA, [6]Auckland Bioengineering Institute, University of Auckland, Auckland 1010, New Zealand, [7]Department ofBioengineering, University of Washington, Seattle, WA 98105, USA and [8]Center for Cell Analysis & Modeling, University of Connecticut School of Medicine, Farmington, CT 06030, USA

[*]These authors contributed equally to this work.
[†]To whom correspondence should be addressed: karr@mssm.edu.

Abstract

Summary: More sophisticated models are needed to address problems in bioscience, synthetic biology, and precision medicine. To help facilitate the collaboration needed for such models, the community developed the Simulation Experiment Description Markup Language (SED-ML), a common format for describing simulations. However, the utility of SED-ML has been hampered by limited support for SED-ML among modeling software tools and by different interpretations of SED-ML among the tools that support the format. To help modelers debug their simulations and to push the community to use SED-ML consistently, we developed a tool for validating SED-ML files. We have used the validator to correct the official SED-ML example files. We plan to use the validator to correct the files in the BioModels database so that they can be simulated. We anticipate that the validator will be a valuable tool for developing more predictive simulations and that the validator will help increase the adoption and interoperability of SED-ML.

Availability: The validator is freely available as a webform, HTTP API, command-line program, and Python package at https://run.biosimulations.org/utils/validate and https://pypi.org/project/biosimulators-utils. The validator is also embedded into interfaces to 11 simulation tools. The source code is openly available as described in the Supplementary data.

Contact: karr@mssm.edu

1 Introduction

Expanded capabilities to predict biological behavior are needed to help engineers design synthetic biological systems and help physicians precisely diagnose and treat disease (Carrera and Covert, 2015; Marucci *et al.*, 2020). Achieving more predictive models will likely require deep collaboration among large teams of modelers, experimentalists, and clinicians (Szigeti *et al.*, 2018; Singla and White, 2021; Waltemath *et al.*, 2011).

To facilitate collaboration, the Computational Modeling in Biology Network (COMBINE; Hucka *et al.*, 2015) developed the Simulation Experiment Description Markup Language (SED-ML; Waltemath *et al.*, 2011), a common format for describing simulations. At its core, SED-ML describes individual simulations of individual models. Initially, SED-ML focused on continuous kinetic models described with CellML (Cuellar *et al.*, 2003) and the Systems Biology Markup Language (SBML; Keating *et al.*, 2020). Recently, we have expanded SED-ML to a broader range of models including spatial, flux balance, qualitative, and rule-based models; a broader range of simulation algorithms such as flux balance analysis (FBA) and asynchronous logical simulation; and additional model languages such as the BioNetGen Language (Faeder *et al.*, 2009), the SBML Flux Balance Constraints (Olivier and Bergmann, 2015) and Qualitative Models (Chaouiya *et al.*, 2013) packages, and Smoldyn (Shaikh *et al.*, 2021).

On top of this core functionality, SED-ML can describe sets of simulations of variants of models, such as an ensemble of stochastic simulations or a parameter scan of a model. In addition, SED-ML can describe how to create tables and figures of simulation results.

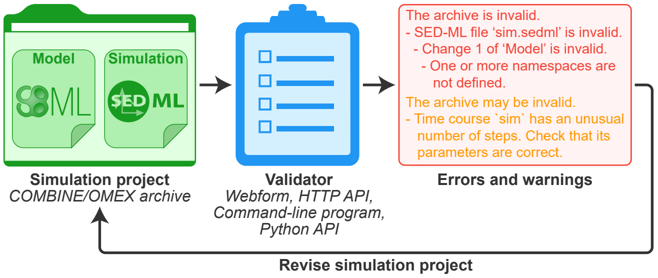
Several tools can create SED-ML files, including web applications such as JWS Online (Peters *et al.*, 2017), RunBioSimulations, and SED-ML Web Tools (Bergmann *et al.*, 2017*b*). SED-ML files can be executed with several tools such as COPASI (Bergmann *et al.*, 2017*a*), iBioSim (Watanabe *et al.*, 2018), OpenCOR (Garny and Hunter, 2015), Tellurium (Choi *et al.*, 2018), and Virtual Cell (Moraru *et al.*, 2008). Furthermore, SED-ML files can be published with repositories such as BioModels (Malik-Sheriff *et al.*, 2020), JWS Online (Peters *et al.*, 2017), and Physiome (Sarwar *et al.*, 2019). More information about these and other SED-ML tools is available at https://sed-ml.org.

However, the utility of SED-ML has been hampered by limited support for SED-ML among modeling software tools and by different interpretations of SED-ML among these tools. For example, we have found that Tellurium can only execute a few of the simulations in BioModels.

To help modelers debug their simulations and to push the community to use SED-ML consistently, we developed a tool that thoroughly validates SED-ML files. The tool is available as a webform, HTTP API, command-line program, and Python API. Here, we articulate how modelers can use the validator, summarize the validation rules the validator checks, describe how the validator communicates issues about SED-ML files, and highlight how we have already used the validator to correct the official SED-ML examples, identify and fill gaps in the SED-ML specifications, and identify bugs in the implementation of SED-ML by multiple software tools. We also outline how we plan to use the validator to correct the SED-ML files in BioModels. The Supplementary data provides more information about the validator and how we are using it to drive convergence around SED-ML.

2 Methods

Because SED-ML files typically describe simulations of external model files encoded in languages such as CellML, NeuroML (Cannon *et al.*, 2014), and SBML, we designed the validator to validate COMBINE archives (Bergmann *et al.*, 2014). COMBINE archives are zip archives that contain one or more SED-ML files, other files that the SED-ML files reference, an OMEX manifest file that summarizes the contents of the archive, and optionally OMEX metadata files (Neal *et al.*, 2020) that capture metadata about the archive and its contents. COMBINE archives can be created with



```
The archive is invalid.
- SED-ML file 'sim.sedml' is invalid.
  - Change 1 of 'Model' is invalid.
    - One or more namespaces are
      not defined.
The archive may be invalid.
- Time course `sim` has an unusual
  number of steps. Check that its
  parameters are correct.
```

Figure 1: The SED-ML validator helps investigators quickly detect errors and other potential problems in SED-ML and model files organized into COMBINE archives.

several tools such as CombineArchiveWeb and RunBioSimulations. More information about these tools is available at https://sed-ml.org.

After creating a COMBINE archive with a SED-ML file, modelers can use the validator via its webform, HTTP API, command-line program, or Python API. Supplementary data S3 and S4 outline how to install and use the validator. To make the validator easy to access, we have also embedded it into standardized interfaces to 11 popular simulation tools (Supplementary data S5.1).

The validator thoroughly checks that COMBINE archives and their contents are consistent with the COMBINE archive, OMEX manifest, OMEX metadata, and SED-ML formats, as well as with the languages of the models in the archive. For example, the validator checks that each reference to a SED-ML element matches the id of an element, that the network of model sources is acyclic, and that each XPath target for each variable of an XML-encoded model matches a single model element. The validator uses LibCellML (https://libcellml.org), LibNeuroML (Vella *et al.*, 2014), and LibSBML (Bornstein *et al.*, 2008) to check that CellML, NeuroML, and SBML models involved in SED-ML files are valid. Supplementary data S2 outlines all of the validation rules that the validator evaluates.

When COMBINE archives are invalid, the validator reports as many errors as can be identified simultaneously, each with contextual information about the element responsible for the error. For example, when the target of a variable of a data generator in a SED-ML file does not match an element of the associated model, the validator provides information about the invalid target and the model that it should match. Supplementary data S4.7 illustrates several example error messages.

The validator also reports warnings about potential mistakes, such as the use of experimental features of SED-ML that few simulation tools support. We implemented warnings for simulations based on common mistakes that we have observed in SED-ML files. We implemented warnings for models using libraries for model languages such as LibSBML.

3 Real-world examples

As a first real-world test, we used the validator to identify and fix issues with the official SED-ML examples (Supplementary data S5.3). The validator alerted us to two common problems with each example, as well as less common issues with several files. The validator also identified files that use a combination of SED-ML elements that the SED-ML specifications do not officially support. This finding prompted us to add a warning for this combination of elements and clarify the description of these examples on the SED-ML website. To ensure these examples remain valid, we also set up an automated action that uses our validator to check these files each time they are changed.

Encouraged by this success, we plan to use our validator to identify and fix issues with the SED-ML files in BioModels (Supplementary data S5.4). We anticipate these corrections will enable these files to be simulated with multiple tools, which will increase the utility of the models in BioModels.

To avoid similar errors in the future, we have also submitted several proposals to clarify the specifications of SED-ML (Supplementary data S5.5) and filed numerous bug reports for several software tools that support SED-ML (Supplementary data S5.6). In addition, we aim to help the BioModels Team incorporate our validator into their curation workflow to ensure that BioModels publishes valid SED-ML files going forward.

4 Discussion

We believe that our validator will be a key resource for debugging simulation experiments and that this work will push the community to use SED-ML consistently. Taken together, we believe these advancements will increase the community's ability to collaborate on simulation experiments, which we anticipate will foster more sophisticated models.

Once the next version of SED-ML (L1V4) is approved, we plan to expand the validator to SED-ML's new features for additional types of observables and plots, data reductions, and model calibration. We also aim to expand the capabilities of the validator to validate additional types of files that could be included in COMBINE archives, such as PETab and Vega files, two emerging formats for model calibration and data visualization.

Funding

This work was supported by the National Institutes of Health [grant number P41EB023912], BMBF LiSyM [grant number 031L0054], and DFG QuaLiPerF [grant number 436883643].

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** (1), 1–9.

Bergmann,F.T. *et al.* (2017*a*) COPASI and its applications in biotechnology. *J. Biotechnol.,* **261**, 215–220.

Bergmann,F.T. *et al.* (2017*b*) SED-ML Web Tools: generate, modify and export standard-compliant simulation studies. *Bioinformatics,* **33** (8), 1253–1254.

Bornstein,B.J. *et al.* (2008) LibSBML: an API library for SBML. *Bioinformatics,* **24** (6), 880–881.

Cannon,R.C. *et al.* (2014) LEMS: a language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2. *Front. Neuroinform.,* **8**, 79.

Carrera,J. and Covert,M.W. (2015) Why build whole-cell models? *Trends Cell. Biol.,* **25** (12), 719–722.

Chaouiya,C. *et al.* (2013) SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.,* **7** (1), 1–15.

Choi,K. *et al.* (2018) Tellurium: an extensible Python-based modeling environment for systems and synthetic biology. *Biosystems,* **171**, 74–79.

Cuellar,A.A. *et al.* (2003) An overview of CellML 1.1, a biological model description language. *Simulation,* **79** (12), 740–747.

Faeder,J.R., Blinov,M.L. and Hlavacek,W.S. (2009) Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol Biol,* **500**, 113–167.

Garny,A. and Hunter,P.J. (2015) OpenCOR: a modular and interoperable approach to computational biology. *Front. Physiol.,* **6**, 26.

Hucka,M. *et al.* (2015) Promoting coordinated development of community-based information standards for modeling in biology: the COMBINE initiative. *Front. Bioeng. Biotechnol.,* **3**, 19.

Keating,S.M. *et al.* (2020) SBML Level 3: an extensible format for the exchange and reuse of biological models. *Mol. Syst. Biol.,* **16** (8), e9110.

Malik-Sheriff,R.S. *et al.* (2020) BioModels–15 years of sharing computational models in life science. *Nucleic Acids Res.,* **48** (D1), D407–D415.

Marucci,L., Barberis,M., Karr,J., Ray,O., Race,P.R., de Souza Andrade,M., Grierson,C., Hoffmann,S.A., Landon,S., Rech,E. *et al.* (2020) Computer-aided whole-cell design: taking a holistic approach by integrating synthetic with systems biology. *Front. Bioeng. Biotechnol.,* **8**, 942.

Moraru,I.I. *et al.* (2008) Virtual Cell modelling and simulation software environment. *IET Syst. Biol.,* **2** (5), 352–362.

Neal,M.L. *et al.* (2020) Open modeling and exchange (OMEX) metadata specification version 1.0. *J. Integr. Bioinform.,* **17** (2-3).

Olivier,B.G. and Bergmann,F.T. (2015) The Systems Biology Markup Language (SBML) Level 3 package: Flux balance Constraints. *J. Integr. Bioinform.,* **12** (2), 660–690.

Peters,M. *et al.* (2017) The JWS Online simulation database. *Bioinformatics,* **33** (10), 1589–1590.

Sarwar,D.M. *et al.* (2019) Model annotation and discovery with the Physiome Model Repository. *BMC Bioinformatics,* **20** (1), 1–10.

Shaikh,B. *et al.* (2021) RunBioSimulations: an extensible web application that simulates a wide range of computational modeling frameworks, algorithms, and formats. *Nucleic Acids Res.,* **49** (W1).

Singla,J. and White,K.L. (2021) A community approach to whole-cell modeling. *Curr. Opin. Syst. Biol.,* **26**, 33–38.

Szigeti,B., Roth,Y.D., Sekar,J.A., Goldberg,A.P., Pochiraju,S.C. and Karr,J.R. (2018) A blueprint for human whole-cell modeling. *Curr. OpiN. Syst. Biol.,* **7**, 8–15.

Vella,M. *et al.* (2014) libNeuroML and PyLEMS: using Python to combine procedural and declarative modeling approaches in computational neuroscience. *Front. Neuroinform.,* **8**, 38.

Waltemath,D. *et al.* (2011) Reproducible computational biology experiments with SED-ML-the Simulation Experiment Description Markup Language. *BMC Syst. Biol.,* **5** (1), 1–10.

Watanabe,L. *et al.* (2018) iBioSim 3: a tool for model-based genetic circuit design. *ACS Syn. Biol.,* **8** (7), 1560–1563.