Bioinformatics, 31(20), 2015, 3383–3386
doi: 10.1093/bioinformatics/btv341
Advance Access Publication Date: 16 June 2015
Applications Note

OXFORD

Systems biology

JSBML 1.0: providing a smorgasbord of options to encode systems biology models

Nicolas Rodriguez[1,2,*,†], Alex Thomas[3,4,*,†], Leandro Watanabe[5], Ibrahim Y. Vazirabad[6], Victor Kofia[7], Harold F. Gómez[8], Florian Mittag[1,9], Jakob Matthes[9], Jan Rudolph[9,10], Finja Wrzodek[1,9], Eugen Netz[9], Alexander Diamantikos[9], Johannes Eichner[9], Roland Keller[9], Clemens Wrzodek[9,11], Sebastian Fröhlich[12,13], Nathan E. Lewis[3,4], Chris J. Myers[5], Nicolas Le Novère[1,2], Bernhard Ø. Palsson[3,4], Michael Hucka[14] and Andreas Dräger[3,9,*]

[1]European Bioinformatics Institute (EBI), Hinxton, UK, [2]Babraham Institute, Babraham Research Campus, Cambridge, UK, [3]University of California, San Diego, La Jolla, CA, USA, [4]Novo Nordisk Foundation Center for Biosustainability, University of California, San Diego, La Jolla, CA, USA, [5]The University of Utah, Salt Lake City, UT, USA, [6]Marquette University, Milwaukee, WI, USA, [7]The University of Toronto, Toronto, ON, Canada, [8]Boston University, Boston, MA, USA, [9]Center for Bioinformatics Tuebingen (ZBIT), University of Tuebingen, Tübingen, Germany, [10]Tel Aviv University, Tel Aviv, Israel, [11]Roche Pharmaceutical Research and Early Development, pRED Informatics, Roche Innovation Center, Penzberg, Germany, [12]Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Gatersleben, Germany, [13]Martin Luther University of Halle-Wittenberg, Halle (Saale), Germany and [14]The California Institute of Technology, Pasadena, CA, USA

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Janet Kelso

Received on December 17, 2014; revised on March 27, 2015; accepted on May 29, 2015

Abstract

Summary: JSBML, the official pure Java programming library for the Systems Biology Markup Language (SBML) format, has evolved with the advent of different modeling formalisms in systems biology and their ability to be exchanged and represented via extensions of SBML. JSBML has matured into a major, active open-source project with contributions from a growing, international team of developers who not only maintain compatibility with SBML, but also drive steady improvements to the Java interface and promote ease-of-use with end users.

Availability and implementation: Source code, binaries and documentation for JSBML can be freely obtained under the terms of the LGPL 2.1 from the website http://sbml.org/Software/JSBML. More information about JSBML can be found in the user guide at http://sbml.org/Software/JSBML/docs/.

Contact: jsbml-development@googlegroups.com or andraeger@eng.ucsd.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The Systems Biology Markup Language (SBML) is a widely used format that enables easy distribution of systems biology data, models and diagrams, and it allows the easy exchange of data and models between a variety of software systems (Hucka *et al.*, 2003; Dräger and Palsson, 2014). Given the wide coverage of the latest SBML version, it is unsurprising that the standard is relatively complex. An appropriate computational architecture greatly simplifies

© The Author 2015. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/31/20/3383/195580 by guest on 10 May 2019

the work that software developers need to do in order to support importing and storing SBML-based information for computational analysis. JSBML (Dräger *et al.*, 2011) is the official, pure Java-based *application programming interface* (API) library for SBML. It enables systems biology information to be expressed in Java data structures patterned after the SBML format for fast access. JSBML 1.0 implements the ability to encode, exchange and use all parts of SBML, up to the current release, SBML Level 3 (L3), including all community-approved (and several prototype) SBML packages for L3, as described later.

One important aspect of SBML is its ability to provide additional capabilities to encode specific types of systems biology models. These capabilities are extensions of the core SBML L3 format and are known as 'packages'. Although many models can be fully represented using only the core set of SBML constructs, these extensions support (i) other model features that cannot be formulated with the SBML core standards and (ii) additional constructs that enable users in specific fields to formulate, interface and use the SBML framework more easily within their modeling approaches. SBML packages enable the ability to build models that encompass several formalisms within one overall framework. Several tools today use JSBML 1.0 in this way and encode multiple types of systems biology models; examples include iBioSim (Madsen *et al.*, 2012), KEGGtranslator (Wrzodek *et al.*, 2011) and GINSim (Gonzalez Gonzales *et al.*, 2006). In addition, because JSBML is built to be an interpreter for SBML, this functionality, represented by a single data structure, can easily be embedded into existing Java programs.

Since its inception, JSBML has fostered a community of active developers who aim to provide regular code updates, provide major and minor bug fixes to releases, and partake in discussions on standards for the systems biology modeling community (COMBINE, Waltemath *et al.*, 2011a). These community interactions have helped improve JSBML substantially since its launch.

2 Results

2.1 Improvements

JSBML's first public release was in 2011 (Dräger *et al.*, 2011) and it has since undergone considerable change and expansion. A major goal of the initial release was to present a software package which differentiated itself from libSBML, JSBML's C++ language counterpart (Bornstein *et al.*, 2008), and its Java-language bindings. Since then, JSBML has maintained compatibility with SBML and libSBML and has introduced extra functionalities.

In addition to support for SBML L3 core and packages, JSBML 1.0 has incorporated improvements to software efficiency. For instance, improvements to internal interfaces now speed up model input/output operations. The different identifier namespaces in SBML are now managed with an IdManager interface which is able to reconcile redundant identifiers among packages. A new Math infix parser is able to mimic the same behavior as the libSBML L3 Math parser and can handle complex mathematical formulas. The manipulation and merging of Units in SBML has also been greatly improved.

Also, several features have been added to JSBML 1.0 to improve end-user convenience and accessibility. For instance, various utility methods have been added that allow users to manipulate JSBML's in-memory data object. Furthermore, user-defined objects can be temporarily added to the JSBML data structure for any model component. XML annotations in SBML are read as XMLNode object instead of Strings, making it easier to manipulate non-standard annotations in SBML. Advanced logging functionalities via the Apache log4j project allows users to monitor JSBML actions.

Finally, JSBML has been better integrated with other software such as Apache Maven, a dependency management tool, BioJava 3 (Prlić *et al.*, 2012), a bioinformatics toolbox, and CellDesigner (Funahashi *et al.*, 2008), a biochemical network modeling and visualization tool.

2.2 Support for SBML packages

2.2.1 Approved packages

At the time of writing, all approved SBML L3 extensions are supported by JSBML 1.0. We describe the support below, and provide insights into the mapping between modeling formalisms and the corresponding JSBML data objects. An abbreviated version of the JSBML class hierarchies are presented in Supplementary Figure S1. The JSBML User Guide has figures that lay out the JSBML class hierarchy for each SBML package, displaying the full capabilities of the JSBML data objects that encode each package. In Table 1, approved SBML packages constitute the first four rows. The *Qualitative Models* package (**qual**, for short) allows species in a model to have non-quantitative or non-continuous levels (Chaouiya *et al.*, 2013). This may manifest as Boolean or discrete values, and is primarily employed in modeling gene regulation, signaling pathways, logical/Boolean networks (Schmulevich *et al.*, 2002), and Petri nets (Breitling *et al.*, 2008). *Flux Balance Constraints* (**fbc**, Olivier and Bergmann, 2013) encodes components for constraints-based modeling (Lewis *et al.*, 2012), which employs a class of models in which the canonical stoichiometric relations between reactions

Table 1. SBML Package status

Name	Id	Description	JSBML support
Qualitative models	qual	Qualitative values for species	Full
Flux balance constraints	fbc	Constraints based parameters	Full
Layout	layout	Network layout topology	Full
Hierarchical model composition	comp	Modular, hierarchical entities	Full
Spatial processes	spatial	Location and geometries	Full
Groups	groups	Grouping elements	Full
Arrays	arrays	Values and entities in arrays	Full
Required elements	req	Required model elements	Full
Distributions	distrib	Model values as statistical distributions	Full
Dynamic structures	dyn	Dynamic model entities	Full
Rendering	render	Network layout style	Full
Multistate and multicomponent species	multi	Rule based modeling	Partial

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/31/20/3383/195580 by guest on 10 May 2019

and metabolites are specified as constraints for mathematical optimization. *Layout* provides the ability to encode graphical information for model diagrams. The structure for this extension mirrors the SBML core hierarchy by introducing graphical counterparts to reactions and species. The fourth approved package, *Hierarchical Model Composition* (**comp**) provides a generic framework to encode models as hierarchical entities in SBML (Smith *et al.*, 2013). JSBML's **comp** implementation provides access to elements within sub-models and interfaces with other models.

2.2.2 Draft packages

Draft specifications are available for the remaining SBML packages; they are encoded in JSBML with varying maturity. JSBML fully supports the current specifications of seven packages whose community approvals are pending: *Spatial Processes* (**spatial**, Schaff *et al.*, 2014) specifies geometric descriptions of biochemical models' components using a cellular coordinate system that can describe non-uniform molecular distributions, diffusive transport and spatially localized reactions; *Groups* (**groups**, Hucka and Smith, 2013) agglomerates SBML model elements and can be linked to annotations and SBO terms (Courtot *et al.*, 2011) to contextualize sets of objects for other programmers and modelers; *Arrays* (**arrays**, Watanabe *et al.*, 2013) extends SBML variables to include arrays of values, thereby representing repeated or regular model structures more efficiently; *Required Elements* (**req**, Smith and Hucka, 2013) allows a model to indicate which components have had their mathematical meanings changed by (e.g.) the use of another SBML package; *Distributions* (**distrib**, Moodie and Smith, 2013) encodes statistical distributions and their sampling; *Dynamic Structures* (**dyn**, Gomez *et al.*, 2014), which supports the definition of dynamical behaviors for model entities; and *Rendering* (**render**, Gauges *et al.*, 2011), used in conjunction with **layout** to provide symbol and style information for diagrams. The last package, *Multistate and Multicomponent Species* (Zhang and Meier-Schellersheim, 2013) is still under development. The JSBML project is committed to support all SBML packages as their specifications come out and are deemed stable by the community.

2.2.3 JSBML and package formation

Some package specifications have been influenced by JSBML development. For example, protocols for validation and flattening of array constructs in the **arrays** package. Prior to the development of **arrays** in JSBML, the specification lacked important validation rules that serve as quality controls for math operations done with **arrays** data structures. JSBML also provided the environment to ensure **arrays** was compatible with other SBML L3 packages, helping to build a more robust specification. Finally, JSBML development of the **arrays** package was used to generate simulatable examples for the SBML L3 **arrays** specification. Therefore, JSBML provides a means to test and implement new package development for future versions of SBML.

3 Conclusions

JSBML version 1.0 marks the maturation of this software library as an essential component for any systems biology pipeline that runs in the Java Virtual Machine, and joins other biological exchange format interpreters (Paxtools, Demir *et al.*, 2013; CellML API, Miller *et al.*, 2010; SED-ML, Waltemath *et al.*, 2011b; SBOL, Galdzicki *et al.*, 2014; libSBGN, van Iersel *et al.*, 2012) to support users' ability to disseminate models in a diverse array of modeling formalisms. In addition, as SBML is updated and novel modeling techniques arise, the active, open-source community behind JSBML will continue to provide a comprehensive, computable interface for systems biology models.

Acknowledgements

The authors are grateful to Meike Aichele, Alexander Dörr, Marine Dumousseau, Akira Funahshi, Stephanie Hoffmann, Sarah R. Müller vom Hagen, Sebastian Nagel, Alexander Peltzer, Simon Schäfer and Andreas Zell.

Funding

A National Institute of Health grant for the continued development of essential SBML software support (NIH, United States, award number GM070923), European Union projects SynSys (FP7 242167) and BBSRC Signalling Institute Strategic Programme (BBS/E/B/000C0419), a National Science Foundation grant (CCF-1218095), a Marie Curie International Outgoing Fellowship within the EU 7th Framework Programme for Research and Technological Development (project AMBiCon, 332020), Google Inc. (Summer of Code 2014), the Federal Ministry of Education and Research (BMBF, Germany) as part of the Virtual Liver Network (grant number 0315756). AT, NEL and BØP acknowledge generous support from the Novo Nordisk Foundation that has been provided to the Center for Biosustainability at the Technical University of Denmark. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our funding agencies. This study was funded in part by NIH HHSN272201000053C.

Conflict of interest: none declared.

References

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

Breitling,R. *et al.* (2008) A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. *Brief. Bioinform.*, **9**, 404–421.

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**, 135.

Courtot,M. *et al.* (2011) Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.*, **7**.

Demir,E. *et al.* (2013) Using biological pathway data with Paxtools. *PLoS Comput. Biol.*, **9**, e1003194.

Dräger,A. and Palsson,BØ. (2014) Improving collaboration by standardization efforts in systems biology. *Frontiers in Bioengineering*, **2**, http://journal.frontiersin.org/article/10.3389/fbioe.2014.00061/.

Dräger,A. *et al.* (2011) JSBML: a flexible Java library for working with SBML. *Bioinformatics*, **27**, 2167–2168.

Funahashi,A. *et al.* (2008) CellDesigner 3.5: a versatile modeling tool for biochemical networks. *Proc. IEEE*, **96**, 1254–1265.

Galdzicki,M. *et al.* (2014) The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature biotechnology*, **32**, 545–550.

Gauges,R. *et al.* (2011) Complementing Layout Information with Render Information in SBML Files. SBML.org (1 January 2015, date last accessed).

Gomez,H. *et al.* (2014) Dynamic Structures, Version 1 Release .1. SBML.org (5 March 2015, date last accessed).

Gonzalez Gonzales,A. *et al.* (2006) GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *BioSystems*, **84**, 91–100.

Hucka,M. and Smith,L.P. (2013) Groups, Version 1 Release .4. SBML.org (1 January 2015, date last accessed).

Hucka,M. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/31/20/3383/195580 by guest on 10 May 2019

Lewis,N.E. *et al.* (2012) Constraining the metabolic genotype-phenotype relationship using a phylogeny of *in silico* methods. *Nat. Rev. Microbiol.*, **10**, 291–305.

Madsen,C. *et al.* (2012) Design and test of genetic circuits using iBioSim. *Design Test Comput. IEEE*, **29**, 32–39.

Miller,A.K. *et al.* (2010) An overview of the CellML API and its implementation. *BMC Bioinformatics*, **11**, 178.

Moodie,S.L. and Smith,L.P.(2013) Distributions, Version .13. SBML.org (5 March 2015, date last accessed).

Olivier,B.G. and Bergmann,F.T. (2013) Flux Balance Constraints, Version 1 Release 1. CO.MBINE.org (1 January 2015, date last accessed).

Prlić,A. *et al.* (2012) BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, **28**, 2693–2695.

Schaff,J.C. *et al.* (2014) Spatial Processes, Version 1 Release .88. SBML.org (5 March 2015, date last accessed).

Shmulevich,I. *et al.* (2002) From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE*, **90**, 1778–1792.

Smith,L.P. and Hucka,M. (2013) Required Elements. SBML.org (1 January 2015, date last accessed).

Smith,L.P. *et al.* (2013) Hierarchical Model Composition, Version 1 Release 3. CO.MBINE.org (1 January 2015, date last accessed).

van Iersel,M.P. *et al.* (2012) Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics*, **28**, 2016–2021.

Waltemath,D. *et al.* (2011a) Meeting report from the fourth meeting of the computational modeling in biology network (COMBINE). *Stand. Genomic Sci.*, **9**, 3.

Waltemath,D. *et al.* (2011b) Reproducible computational biology experiments with SED-ML-the simulation experiment description markup language. *BMC Syst. Biol.*, **5**, 198.

Watanabe,L. *et al.* (2013) Arrays, Version 1 Release .1a. SBML.org (1 January 2015, date last accessed).

Wrzodek,C. *et al.* (2011) KEGGtranslator: visualizing and converting the KEGG PATHWAY database to various formats. *Bioinformatics*, **27**, 2314–2315.

Zhang,F. and Meier-Schellersheim,M. (2013) Multistate, Multicomponent, and Multicompartment Species, Version 1 Release .1. SBML.org (1 January 2015, date last accessed).

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/31/20/3383/195580 by guest on 10 May 2019