doi:10.20944/preprints202012.0296.v1

The Systems Biology Simulation Core Library

Hemil Panchiwala[1†], Shalin Shah[2,3†], Hannes Planatscher[4], Mykola Zakharchuk[5], Matthias König[6], and Andreas Dräger[5,7,8,9,*]

[1]Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee, India - 247667
[2]Department of Electrical and Computer Engineering, Duke University, Durham, NC 27701, United States of America
[3]Bloomberg LP, New York, NY 10022, United States of America
[4]Signatope GmbH, 72770 Reutlingen, Germany
[5]Department of Computer Science, University of Tübingen, 72076 Tübingen, Germany
[6]Institute for Theoretical Biology, Humboldt University of Berlin, 10115 Berlin, Germany
[7]Computational Systems Biology of Infections and Antimicrobial-Resistant Pathogens, Institute for Bioinformatics and Medical Informatics (IBMI), University of Tübingen, 72076 Tübingen, Germany
[8]German Center for Infection Research (DZIF), partner site Tübingen, Germany
[9]Cluster of Excellence 'Controlling Microbes to Fight Infections,' University of Tübingen, Tübingen, Germany
[†]These authors contributed equally to the manuscript.
[*]Corresponding author: draeger@informatik.uni-tuebingen.de

ABSTRACT

Summary: Studying biological systems generally relies on computational modeling and simulation, e.g., for model-driven discovery and hypothesis testing. Progress in standardization efforts led to the development of interrelated file formats to exchange and reuse models in systems biology, such as SBML, the Simulation Experiment Description Markup Language (SED-ML), or the Open Modeling EXchange format (OMEX). Conducting simulation experiments based on these formats requires efficient and reusable implementations to make them accessible to the broader scientific community and to ensure the reproducibility of the results. The Systems Biology Simulation Core Library (SBSCL) provides interpreters and solvers for these standards as a versatile open-source API in Java™. The library simulates even complex bio-models and supports deterministic Ordinary Differential Equations (ODEs); Stochastic Differential Equations (SDEs); constraint-based analyses; recent SBML and SED-ML versions; exchange of results, and visualization of *in silico* experiments; open modeling exchange formats (COMBINE archives); hierarchically structured models; and compatibility with standard testing systems, including the Systems Biology Test Suite and published models from the BioModels and BiGG databases.
Availability and implementation: SBSCL is freely available at https://draeger-lab.github.io/SBSCL/.

Keywords: Systems Biology, Numerical Solver, Java™, API Library, SBML, SED-ML, OMEX, Constraint-Based Modeling, Stochastic Simulation, Ordinary Differential Equation Systems

1 Introduction

The Systems Biology Simulation Core Library (SBSCL) is an open-source, cross-platform pure Java™ programming library that numerically solves systems biology models in multiple mathematical frameworks. A popular file format for representing computational models in a standard way and facilitating the exchange of models between different tools is the Systems Biology Markup Language (SBML)[7]. SBML encodes biological models in a declarative form. The Simulation Experiment Description Markup Language (SED-ML) format defines a workflow of simulation experiments. Using SED-ML and SBML in combination enables the reproducibility of typical model workflows in *in-silico* experiments, including the choice of interpretation framework and the postprocessing of the results[15]. SBSCL interprets the SBML models using the JSBML library[4,13] and simulates them according to dedicated API calls. Alternatively, it extracts an *in silico* experimental configuration from SED-ML to simulate the SBML models. To this end, SBSCL implements and ships several solvers for a wide range of mathematical frameworks, including ODEs[8], SDEs[6], and constraint-based analysis. SBSCL is designed as a lightweight API and intended for use as a simulation backend within end-user software, such as SBMLsimulator[2,3]. This article introduces the SBSCL library, especially the new features introduced in version 2.0, along with a brief description of all other capabilities that are pictorially summarized in figure 1 on the following page.

2 Description

Differential equation solver: The most fundamental feature of SBSCL is simulating ODEs. Version 2.0 adds interpreters and solvers for SDEs to support the latest SBML standards. SBSCL efficiently implements three deterministic numerical solvers[8], namely, Rosenbrock, Euler, and Runge-Kutta, as well as three stochastic solvers, namely, Gillespie, Gibson-Bruck, and Tau-Leaping[6].

Constrained optimization solver: SBML Level 3[7] combined with the `fbc` package[12] added support for constrained-based models and their analysis. Typically, Flux Balance Analysis (FBA) is used for such time-invariant steady-state

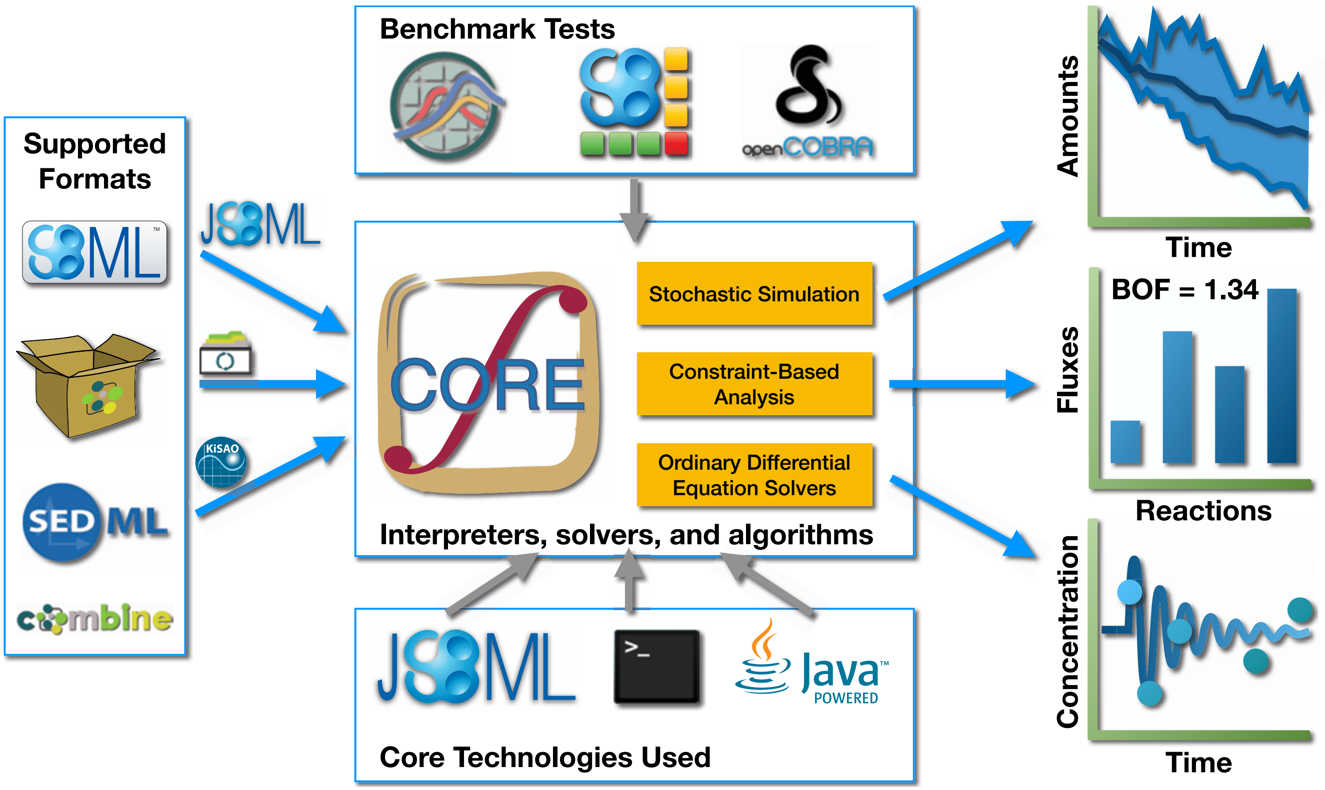

Figure 1. The capabilities of the Systems Biology Simulation Core Library (SBSCL) as an overview. Supported input model definitions include SBML, possibly with an experiment configuration file (SED-ML) or bundled in a package file (OMEX). The model is parsed using the JSBML library, and solutions are numerically computed for the corresponding ODE or SDE over time, following the specified constraints and algorithm (e.g., Rosenbrock, Euler, Gillespie) or via linear programming. Once the simulation completes, the model results are reported either graphically using a line plot or tabular form. The results can be exported to formats such as Comma-Separated Values (CSV) for downstream use. For testing the library, its implementation, its robustness, reliability, and efficient reproducibility of the results, open model collections such as BiGG[9,11] and BioModels[10] are utilized, which comprise several hundred SBML models and their SED-ML configurations.

simulations. SBSCL performs FBA on SBML models using the SCPSolver (`http://www.scpsolver.org`), a linear programming API with support for various solver backends. This lightweight abstraction allows users to define model constraints and an objective function and solve the corresponding optimization problem.

Result tables and plots: Since viewing is an essential aspect of understanding the results of a simulation experiment, SBSCL provides experiment output in graphical and tabular form, which it can export in conventional formats such as CSV.

Archival format support: Working towards exchangeability and reproducibility, SBSCL version 2.0 supports the OMEX format as input. These archive files contain the information on running simulation experiments based on SBML and SED-ML[1]. SBSCL uses the COMBINE Archive Simulation Experiment Management for Systems Biology (SEMS) package to read and extract the required information from the OMEX files.

Hierarchical model simulations: The SBML extension package `comp`[14] enables encoding complex and coupled biological systems that can be distributed or hierarchically structured. SBSCL version 2.0 efficiently supports simulation of this addition, including the automatic assembly of models from multiple and possibly remote input files.

Tests against benchmark suites: A crucial part of implementing new features is providing robust testing of the added functionality and use-cases. SBSCL tests all newly added features against the SBML Test Suite in a continuous integration approach. SBSCL provides full testing support against the genome-scale models from the BiGG Models database with COBRApy[5] and kinetic models from the BioModels database.

3 Conclusion

The open-source library SBSCL simulates complex biological models in various frameworks specified in SBML format, optionally together with their *in silico* experiment definition

SED-ML file or wrapped within OMEX archives. Benchmarks of SBSCL using the SBML Test Suite and a broad range of published models from relevant databases ensure its correctness and reliability. With the support for exciting new features such as constraint-based model optimization, hierarchical model decomposition, stochastic algorithms, archival input formats, this lightweight library is well suited as a simulation engine within any software with support for the Java Virtual Machine, e.g., Kotlin, Scala, or Groovy. The SBSCL project aims to provide a high-quality open-source simulation library to the scientific community to push frontiers and reproducibility in biology and related fields.

Acknowledgments

The authors acknowledge contributions by Nicolas Rodriguez, Alexander Dörr, Roland Keller, and the Harvey Mudd College for the ODEToolkit.

Funding: The National Resource for Network Biology (NRNB) and Google Inc. supported this work as part of their summer of code programs (GSoC). AD was funded by the German Center for Infection Research (DZIF), grant № 8020708703, and supported by infrastructural funding from the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation), Cluster of Excellence EXC 2124 Controlling Microbes to Fight Infections. MK is supported by the Federal Ministry of Education and Research (BMBF, Germany) within the research network Systems Medicine of the Liver (LiSyM, grant № 031L0054) and by the DFG within the Research Unit Programme FOR 5151 “QuaLiPerF (Quantifying Liver Perfusion–Function Relationship in Complex Resection—A Systems Medicine Approach)” by grant № 436883643.

Conflict of interest: None

Abbreviations

API	Application Programming Interface
BiGG	Biochemically, Genomically, Genetically structured
BMBF	Federal Ministry of Education and Research
COMBINE	Computational Modeling of Biological Networks
`comp`	SBML extension package for Hierarchical Model Composition
CSV	Comma-Separated Values
DFG	*Deutsche Forschungsgemeinschaft*, German Research Foundation
DZIF	German Center for Infection Research
FBA	Flux Balance Analysis
`fbc`	SBML extension package for Flux Balance Constraints
GSoC	Google Summer of Code
NRNB	National Resource for Network Biology
ODE	Ordinary Differential Equation
OMEX	Open Modeling EXchange format
SBML	Systems Biology Markup Language
SBSCL	Systems Biology Simulation Core Library
SDE	Stochastic Differential Equation
SED-ML	Simulation Experiment Description Markup Language
SEMS	Simulation Experiment Management for Systems Biology

References

1. Bergmann, F.T., Adams, R. et al. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics*, 15:369, December 2014 doi: 10.1186/s12859-014-0369-z.
2. Buchweitz, L.F., Yurkovich, J.T. et al. Visualizing metabolic network dynamics through time-series metabolomic data. *BMC Bioinformatics*, 21(1):130, April 2020 doi: 10.1186/s12859-020-3415-z.
3. Dörr, A., Keller, R. et al. SBMLsimulator: a Java tool for model simulation and parameter estimation in systems biology. *Computation*, 2(4):246–257, December 2014 doi: 10.3390/computation2040246.
4. Dräger, A., Rodriguez, N. et al. JSBML: a flexible Java library for working with SBML. *Bioinformatics*, 27(15):2167–2168, June 2011 doi: 10.1093/bioinformatics/btr361.
5. Ebrahim, A., Lerman, J.A. et al. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Systems Biology*, 7:74, August 2013 doi: 10.1186/1752-0509-7-74.
6. Erhard, F., Friedel, C.C. et al. FERN – a Java framework for stochastic simulation and evaluation of reaction networks. *BMC Bioinformatics*, 9:356, August 2008 doi: 10.1186/1471-2105-9-356.
7. Keating, S.M., Waltemath, D. et al. SBML Level 3: an extensible format for the exchange and reuse of biological models. *Mol. Syst. Biol.*, 16(8):e9110, August 2020 doi: 10.15252/msb.20199110.
8. Keller, R., Dörr, A. et al. The systems biology simulation core algorithm. *BMC Systems Biology*, 7:55, July 2013 doi: 10.1186/1752-0509-7-55.
9. King, Z.A., Lu, J.S. et al. BiGG Models: A platform for integrating, standardizing, and sharing genome-scale models. *Nucleic Acids Research*, October 2015 doi: 10.1093/nar/gkv1049.
10. Malik-Sheriff, R.S., Glont, M. et al. BioModels—15 years of sharing computational models in life science. *Nucleic Acids Research*, 48:D407–D415, January 2020 doi: 10.1093/nar/gkz1055.
11. Norsigian, C.J., Pusarla, N. et al. BiGG Models 2020: multi-strain genome-scale models and expansion across the phylogenetic tree. *Nucleic Acids Res.*, 11 2019 doi: 10.1093/nar/gkz1054, gkz1054.
12. Olivier, B.G. and Bergmann, F.T. SBML Level 3 Package: Flux Balance Constraints version 2. *Journal of Integrative Bioinformatics*, 15, March 2018 doi: 10.1515/jib-2017-0082.
13. Rodriguez, N., Thomas, A. et al. JSBML 1.0: providing a smorgasbord of options to encode systems biology models. *Bioinformatics*, June 2015 doi: 10.1093/bioinformatics/btv341.
14. Smith, L.P., Hucka, M. et al. SBML Level 3 package: Hierarchical Model Composition, Version 1 Release 3. *Journal of Integrative Bioinformatics*, 12:268, September 2015 doi: 10.2390/biecoll-jib-2015-268.
15. Waltemath, D., Adams, R. et al. Reproducible computational biology experiments with SED-ML-the simulation experiment description markup language. *BMC Systems Biology*, 5(1):1–10, 2011 doi: 10.1186/1752-0509-5-198.