BIOINFORMATICS Bioinformatics Advance Access published March 9, 2006

The SBML ODE Solver Library: a native API for symbolic and fast numerical analysis of reaction networks

Rainer Machné [a], Andrew Finney [b], Stefan Müller [c], James Lu [c], Stefanie Widder [a], Christoph Flamm [a,d]

[a]Theoretical Biochemistry Group, Institute for Theoretical Chemistry, Univ. of Vienna, Währingerstr. 17, 1090 Vienna, Austria, [b]Physiomics PLC, Magdalen Centre, Oxford Science Park, Oxford, OX4 4GA, UK, [c]Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Altenbergerstrasse 69, 4040 Linz, Austria, [d]Bioinformatics, Institute for Computer Science, Univ. of Leipzig, Härtelstr. 16-18, 04107 Leipzig, Germany

Associate Editor: Martin Bishop

ABSTRACT

The *SBML ODE Solver Library* (*SOSlib*) is a programming library for symbolic and numerical analysis of chemical reaction network models encoded in the Systems Biology Markup Language (*SBML*). It is written in *ISO C* and distributed under the open source *LGPL* license. The package employs *libSBML* structures for formula representation and associated functions to construct a system of ordinary differential equations (*ODE*s), their Jacobian matrix and other derivatives. *SUNDIALS*' *CVODES* is incorporated for numerical integration and sensitivity analysis. Preliminary benchmarking results give a rough overview on the behavior of different tools and are discussed in the supplementary material. The native *API* provides fine-grained interfaces to all internal data structures, symbolic operations and numerical routines, enabling the construction of very efficient analytic applications and hybrid or multi-scale solvers with interfaces to *SBML* and non *SBML* data sources. Optional modules based on *XMGrace* and *Graphviz* allow quick inspection of structure and dynamics.

Availability: www.tbi.univie.ac.at/∼raim/odeSolver/

Contact: {raim, xtof}@tbi.univie.ac.at

1 INTRODUCTION

Mathematical modeling of (bio)chemical reaction networks involves a variety of techniques and theories and has long been applied for many purposes in research and technology. The need for exchange of models between different computational tools motivated collaborative efforts to develop standard formats for describing the common chemical reaction networks underlying the differing derived mathematical descriptions. Of the two *XML* based standards *SBML* (Hucka *et al*., 2003) and *CellML* (Lloyd *et al*., 2004), the former is supported by a growing number of applications and an official programming library, *libSBML* (http://sbml.org/software/libsbml/). While available tools (see http://www.sbml.org) cover a variety of methods to edit and analyze reaction networks and their dynamics and/or structure, they are mostly designed as platform specific standalone tools, accessible mostly via complex graphical user interfaces. In contrast, *SOSlib* combines *libSBML* with the *SUNDIALS* package (http://www.llnl.gov/CASC/sundials/) to provide a detailed *API* (application program interface) to both a derived *ODE* system and various derivatives thereof (e.g. the Jacobian matrix) and to efficient integration and sensitivity analysis routines. The fine-grained interfaces allow to access integration routines at all levels, e.g. to operate on parameter and variable values during integration or to incorporate external data. All functionalities of the library are implemented in several well-documented example programs and a simple command-line application with additional visualization modules.

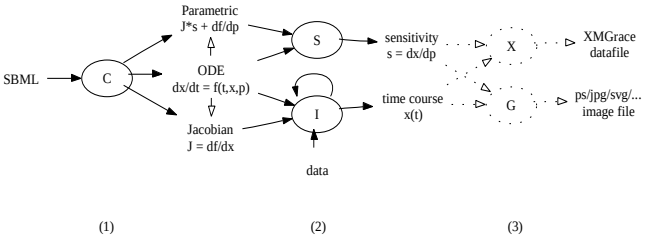

Fig. 1. Basic architecture of the *SOSlib*. C: construction of analytic structures; I: numerical integration; S: sensitivity analysis; G: graph drawing; X: result visualization. See text for details about the functional levels (1)-(3). The API includes detailed interfaces to all three levels.

2 METHODS

The *SOSlib* is a straightforward integration of the features of *libSBML*, the official library for parsing and editing *SBML*, with *CVODES*' methods for solving stiff and non-stiff *ODE* systems and their parameter sensitivities. Figure 1 depicts the basic architecture, which can be outlined as follows:

(1) Construction of an *ODE* system ($dx/dt = f(x, p, t)$, where p are parameters of the system) from a reaction network follows the usual procedure, as described in many text books. *SBML*'s 'kinetic law' construct represents reaction kinetics in *(item or mole)/second* instead of the usual *mole/(liter*second)*, allowing a very simple handling of multi-compartment systems. The *SBML* 'rate rule' construct enables representation of *ODE*s directly, and *SOSlib* actually constructs a derived *SBML* model, consisting only of rate rules (*ODE*s) for the system's variables. Simple recursive functions for formula evaluation, symbolic differentiation and simplification based on *libSBML*'s abstract syntax tree (*AST*) representation of formulae further allow to construct the Jacobian matrix ($J : \delta f/\delta x$) and differentiation with respect to parameters of the ODE system ($P : \delta f/\delta p$). Differentiation is implemented for all formulae representable in *AST*. Structures in f that can't be differentiated with respect to y will produce messages in the returned formula for df/dy as well as in *SOSlib*'s error management system.

(2) The constructed *ODE* system and its derivative J are then used to initiate numerical integration by *CVODE*'s implementation of the *BDF* (backward differentiation formula) or the *Adams-Moulton* (*AM*) method

Table 1. Benchmarking of the *SOSlib* and other *SBML* Solvers

Biomodel DB Id	9	14	33	22	repress.
Type	stable	stable	stable	oscil.	oscil./stiff
NEQ/Time	22/150	86/300	28/60	10/2000	6/10,000
Dizzy 1.11.1	15,499	12,711	2,634	19,350	6,369
Jarnac 2.16n	344	14,531	1,157	5,843	4,516
SBMLToolbox	188	920	302	5,554	6,681
SOSlib 1.6	234	515	171	562	1,062
Copasi 4.0.b15	156	4,062	109	1,437	500

CPU times of single runs in milliseconds for ODE construction and numerical integration on a Pentium 4 CPU with 3.4 *GHz* and 1GB RAM. See supplementary material for details. The column numbers are the models' *ID*s at the BioModels DB (Le Novère *et al*., 2006), except for 'repress.' ('repressilator') which was taken from Müller *et al*., 2005. NEQ: number of ODEs. Time: end time of integration in the model's built in default unit. Type: refers to the model's dynamic behavior.

with *Newton* or *Functional iteration* to calculate $x(t)$ for a requested series of time points. *BDF* and *AM* methods are used for stiff and non-stiff systems, respectively. Changing variables $x(t)$ at any time point by either event assignments (see below) or by calling applications simply requires reinitialization of the integrator structures with new initial values. *SOSlib* incorporates *CVODES*' methods for forward sensitivity analysis using the parametric derivatives in P to calculate $\delta x(t)/\delta p$. The performance of the methods for iteration as well as for forward sensitivity analysis (*staggered direct*, *simultaneous* or *staggered corrector*) depend on the numerical properties of the specific system. When construction of J or P fails due to undifferentiable structures in *ODE*, internal approximation routines of *CVODES* are employed. Both, numerical integration and sensitivity analysis are compatible with online variable manipulation. *SBML* event triggers are evaluated at every time step and executed if fired. The flaws of this approach are that the accuracy of event detection depends on the chosen time step and that the order of events fired at the same time step is not further resolved.

(3) Two optional modules support visualization of a model's structure and dynamics. Time courses of concentrations, rates, reaction fluxes, and Jacobian matrix values can be directly visualized in *XMGrace* (http://plasmagate.weizmann.ac.il/Grace/). The *Graphviz* library is employed for graph drawing (http://www.graphviz.org). Besides the usual bipartite reaction network, a *species interaction* and a *parameter dependency* graph based on J and P respectively, proved useful for visual exploration of dominating feedback cycles and parameter dependencies.

3 ACCURACY, SCOPE AND APPLICATIONS

3.1 Numerical Analysis

SOSlib's integration routines have been tested with the official *SBML Semantic Test Suite*. All of the tests without algebraic rules, events or delays were successfully solved. We have benchmarked the performance of *SOSlib* and four other *SBML ODE* solvers, namely *Jarnac*, *Copasi*, *Dizzy* and the *SBMLToolbox* for *Matlab*, using 4 models from the BioModels DB (Le Novère *et al*., 2006) and a variant of the *repressilator* model in a stiff parameter regime (Müller *et al*., 2005). Models 22 and the *repressilator* are oscillatory models, the others approach a steady state. *SOSlib* performed especially well with the high-dimensional model (DB Id 14) and with the oscillatory models. The benchmark results provide first hints on the performance implications of the implementation approaches used by the selected tools. The results also show that *SOSlib* performs better than the majority of those tools (see table 1). References for tools and models, and a short discussion of benchmarking difficulties and results are available as supplementary material.

Limitations: *SOSlib* (1.6.0) does not support user defined units or unit conversions. Models with delays or algebraic rules can currently not be handled. The event detection and execution is not generally valid and has to be used with care (see *Methods*).

3.2 Symbolic Analysis & Hybrid Systems

All formulae in *SBML*, *ODE*, J and P can be retrieved as *libSBML AST* structures, evaluated with current data or further differentiated with respect to arbitrary variables. This functionality for symbolic operations opens *SBML* models for further analytic treatment.

The fine-grained interface to numerical routines directly enables efficient multi-scale modeling or integration into hybrid (continuous + discrete) solvers. Independent solver instances can 'communicate' between time steps via straight-forward functions to set all parameters and variables, including the next time step of integration. Furthermore, an application can provide functions that evaluate external data from e.g. experimental measurements or other solvers.

4 DISCUSSION

While systems biology has gained much attention only recently, it's current methods are in large parts old and well-founded on mathematical and biochemical theories. We follow the spirit of *SBML* and *libSBML* to provide application developers with detailed interfaces to already existing standard methods under a very liberal licensing policy. In other words: we are not trying to 'reinvent the wheel', but offer its best possible implementation to enable rapid scientific progress and unrestricted further development in this field. *SOSlib* uniquely provides a detailed native *API*, independent of any *GUI* or scripting environment, that allows access to all the components of a deterministic reaction network simulator enabling scientists to construct efficient applications that are tailored to their research needs. We are not aware of any other open source *SBML* analysis package that offers the time course sensitivity analysis provided by *SOSlib* through *CVODES*. The next releases (during 2006) will include exact event handling and extend the *CVODES* interface to provide adjoint sensitivities and sophisticated parameter identification routines that employ inverse methods. *SUNDIALS*'s *IDA* solver for differential algebraic equation systems will be interfaced to provide integration for models with arbitrary algebraic rules. Bifurcation and feedback analysis would be obvious next steps.

ACKNOWLEDGMENT

This work was supported by the WWTF, project number MA05.
RM gratefully acknowledges financial support by his parents.

REFERENCES

S. Cohen and A. Hindmarsh (1996) CVODE, a stiff/nonstiff ODE solver in C. *Computers in Physics*, 10(2):138-143.

M. Hucka *et. al*. (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524-531.

N. Le Novère *et. al*. (2006) BioModels Database: A Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems. *Nucleic Acids Res., in the press*

CM. Lloyd, MD. Halstead, and PF. Nielsen. (2004) CellML: its future, present and past. *Prog Biophys Mol Biol*, 85(2-3):433-450.

S. Müller, J. Hofbauer, L. Endler, C. Flamm, S. Widder, P. Schuster (2005). A Generalized Model of the Repressilator. submitted to *Bull Math Bio.*