BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–2

SBML ODE Solver Library: a command-line tool and library for numerical analysis of reaction networks

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

[a]Theoretical Biochemistry Group, Institute for Theoretical Chemistry, University of Vienna, Währingerstrasse 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, University of Leipzig, Härtelstrasse 16-18, 04107 Leipzig, Germany

ABSTRACT

The SBML ODE Solver Library (SOSlib) is a command-line oriented tool and programming library for construction and numerical integration of a system of ordinary differential equations (ODE) from a chemical reaction network encoded in the Systems Biology Markup Language (SBML). It is written in ANSI C, provides bindings for a variety of scripting languages and is distributed under LGPL license. The package uses the SBML library libSBML for parsing SBML models and construction of ODE systems, and CVODE for numerical integration. Optional data visualization modules based on XMGrace and Graphviz allow a quick inspection of the model's structure and dynamics, providing a transparent and direct interface to SBML models, which is especially useful for educational purposes.

Availability: http://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. Diverse but potentially complementary approaches have been taken to analyze networks of chemical reactions, roughly dividable in 'dynamical' and 'structural' analysis. The need for exchange of models between the different available computational tools motivated collaborative efforts to develop a standard format for describing the common chemical reaction networks underlying the differing derived mathematical descriptions. Of two XML based community standards, SBML (Hucka *et al*., 2003) and CellML (Lloyd *et al*., 2004), the former is supported by a growing number of sophisticated applications and an official programming library, Ben Bornstein's libSBML (http://www.sbml.org/software/libsbml/). The available tools (see SBML website at http://www.sbml.org) cover a variety of methods to edit and analyze reaction networks and their dynamics and/or structure. However, they are mostly designed as platform specific standalone tools whose functionality is only accessible via more or less complex user interfaces. Through it's easy-to-use and stripped down functionality, the SOSlib offers itself both as a stand-alone tool for quick exploration of a system's structure and dynamics and as a simple and reliable programming library, that provides a powerful platform-independent integration back-end for higher-level SBML analysis or visualization tools. Thus the SOSlib is targeted at biomathematicians, 'command-line friendly' biochemists and biologists, and at application developers, respectively. As SBML is able to represent not only reaction networks but arbitrary ODE systems, the tool can be considered as a general ODE solver, or as an interface to the established and well tested CVODE environment for solving non-stiff and stiff ODE systems (Cohen and Hindmarsh, 1996), as distributed within the SUNDIALS package (http://www.llnl.gov/CASC/sundials/main.html). The fine grained interface allows to access integration routines at all levels.

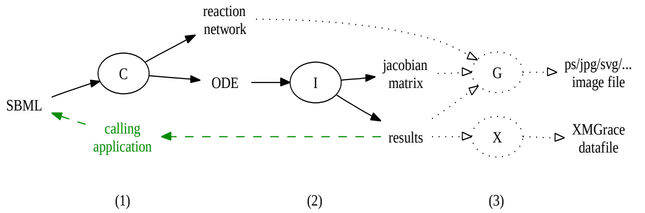

Fig. 1. Basic architecture of the SOSlib. See text for details about the functional levels (1)-(3). The API allows external applications to interface at all three levels. Dotted nodes represent optional modules for data visualization.

2 METHODS

The SOSlib is a straightforward integration of the features of libSBML, the official library for parsing and editing SBML, with CVODE's sophisticated methods for solving stiff and non-stiff ODE systems. CVODE is also used in the classical tool SCAMP (Sauro, 1993). Figure 1 depicts the basic architecture, which can be outlined as follows:

(1) Construction of ODE systems from reaction networks follows the usual procedure, as described in many text books. A main difference is the SBML kinetic law, which represents reaction kinetics in units of 'substance/time', instead of the classical rate law descriptions in 'concentration/time'. This allows a very simple handling of multi-compartment systems. The SBML 'rate rule' construct allows to represent ODEs directly, and the SOSlib actually constructs a second model, consisting only of rate rules (ODEs) for the system's variables. This conversion of a general SBML model to a defined subset of SBML takes care of all other special constructs of SBML, such as user defined functions, parameter assignments, discrete events, delays and algebraic rules. The latter three constructs, events, delays and algebraic rules, cannot be treated within the realm of ODE systems and can currently

*to whom correspondence should be addressed

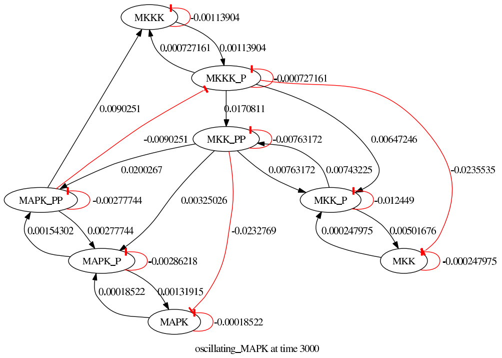

Fig. 2. Jacobian matrix graph of the oscillating MAP kinase model by Kholodenko, 2000, generated by the `SBML ODE Solver`.

not be solved (exactly) by the `SOSlib`.

(2) `libSBML`'s abstract syntax tree (`AST`) represents formulas in their correct precedence, encoded in a tree structure. This allowed the design of simple recursive functions for formula evaluation, symbolic differentiation and simplification. The constructed `ODE` system is then used to initiate numerical integration by `CVODE`'s implementation of variable-coefficient forms of the BDF method (backward differentiation formula) and Newton iteration for solving both stiff and non-stiff systems. `CVODE`'s methods furthermore require a function that calculates the current values of the Jacobian matrix. The Jacobian can either be derived by symbolic differentiation, or approximated internally by `CVODE`. The `SOSlib` allows the user or the interfacing program to set all of `CVODE`'s integration parameters, such as absolute and relative error tolerances.

3 RESULTS

3.1 Accuracy and Scope

The `SOSlib` has been extensively tested with the official `SBML Semantic Test Suite`. All of the tests without algebraic rules, events or delays (104 of 113 test models) were successfully integrated. The test suite includes models at the extremes of low numerical values, and they were solved without problem. Detailed results of this test run, and instruction how to reproduce the tests are distributed with the source code.

Discrete events can be handled in principle by stopping the integration upon firing of an event's trigger and starting a new integration run, with new values, defined by the event's assignment rules. This approach is currently implemented, but the accuracy of event detection depends on the chosen time interval of integration steps and should thus be used with care.

3.2 Data Visualization

(3) Two optional modules, that depend on additional libraries, are used to support visual exploration of the model's structure and dynamics. First, time courses of concentrations, rates, reaction fluxes, and Jacobian matrix values can be directly visualized in `XMGrace` (http://plasma-gate.weizmann.ac.il/Grace/). Second, `SOSlib` uses algorithms from the Graphviz library for graph drawing (http://www.graphviz.org). Besides the usual bipartite reaction network graph, a `species interaction graph`, based on the non-zero entries of the Jacobian matrix, can be constructed. This graph representation of the Jacobian proved very practical for visual exploration of the dynamic regulation of small reaction networks, e.g. to get a first impression of possible and relevant positive or negative feedback cycles within a reaction network. Figure 2 shows such a graph for the MAPK pathway's phosphorylation cascade with negative feedback, leading to oscillations (Kholodenko, 2000). This model has been obtained from the official `SBML` model repository at http://www.sbml.org/models/. Finally, an 'interactive mode' allows easy access to the internals of an `SBML` model, motivates to tinker with parameters and initial values, and is therefore especially suited for educational purposes.

4 DISCUSSION

Several groups have already incorporated the `SOSlib` as a backend for numerical integration within higher-level `SBML` applications, such as the `CellDesigner` (Funahashi *et al.*, 2003). Their experience and suggestions had and will have a strong impact on further development. We expect the `SOSlib` to become part of a set of additional layers extending the basic functionality of `libSBML`, providing application developers in the fields of bioinformatics and systems biology with well-founded methodology for an integrated analysis of biological reaction networks. The functionality for symbolic operations opens `SBML` models (and arbitrary `ODE` systems) for analytical treatment. We are currently developing a sophisticated parameter estimation application that employs inverse methods. This extension will allow automatic estimation of reasonable values for partially under-determined models. Sophisticated methods of feedback analysis and derivation of higher-order models of biological function (e.g. Thomas and Kaufman, 2001) could also greatly benefit from the symbolic computation.

The `SUNDIALS` package additionally features the `IDA` and `CVODES` solvers. `IDA` enables the integration of differential algebraic equation (`DAE`) systems which will allow to handle `SBML` models with algebraic rules. `CVODES` includes sensitivity analysis in the solution of `ODE` systems. `DAE` and sensitivity analysis functionality provided in large part by `SUNDIALS` will be made available in the next release of `SOSlib`, which can be expected by the end of 2005. Support for `ODE`s with delays, the exact detection of events and integration with stochastic tools to create a hybrid simulation engine ranks high in our priorities.

ACKNOWLEDGEMENT

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.

A. Funahashi , N. Tanimura, M. Morohashi, and H. Kitano (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, 1:159-162.

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.

BN. Kholodenko (2000) Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. *Eur J Biochem*, 267(6):1583-1588.

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

H. Sauro (1993) SCAMP: a general-purpose simulator and metabolic control analysis program. *Comput Appl Biosci.*, 9(4):441-50.

R. Thomas and M. Kaufman (2001) Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11(1):180-195.