

Systems biology

Tools for the SBML Community

Colin S. Gillespie^{1,*}, Darren J. Wilkinson¹, Carole J. Proctor², Daryl P. Shanley², Richard J. Boys¹ and Thomas B. L. Kirkwood²

¹School of Mathematics and Statistics, University of Newcastle, Newcastle upon Tyne, NE1 7RU, UK and

²Henry Wellcome Laboratory for Biogerontology Research, School of Clinical and Medical Sciences, Gerontology, University of Newcastle, Newcastle upon Tyne, NE4 6BE, UK

Received on September 22, 2005; revised on December 7, 2005; accepted on January 4, 2006

Advance Access publication January 12, 2006

Associate Editor: Thomas Lengauer

ABSTRACT

Motivation: SBML is quickly becoming the standard format to exchange biochemical models. The tools presented in this paper are loosely-coupled, and are intended to be incorporated into SBML aware applications. The rationale for this is to reduce the amount of repeated work carried out within the community and to create tools that offer a greater number of features to the end-user.

Availability: All tools described are available from <http://www.basis.ncl.ac.uk/software> and are licensed under GNU General Public License.

Contact: c.gillespie@ncl.ac.uk

INTRODUCTION

In recent years an increasing emphasis has been placed on modelling biochemical networks with the aim of informing biologists about their complex functioning and to aid in designing experimental strategies. There is no unique correct method for modelling; rather there are a large number of techniques and theories which provide tools for exploring model systems. Owing to the numerous modelling strategies available it is unlikely that any one group can develop a set of tools to carry out all possible analyses. Hence, an effort has been made to develop a standard format, the Systems Biology Mark-up Language (SBML) (Hucka *et al.*, 2003), which captures all of the necessary information relating to a model in a XML format. This allows the model to be imported into and exported out of a variety of tools. The SBML is quickly becoming the *lingua franca* for development and sharing models of biochemical networks. Currently, there are a large number of tools that support SBML, however they tend to be designed as largely stand-alone, platform-specific applications.

In this paper we describe a new open source collection of modelling tools that can be combined with existing applications to increase usability. Although the tools can be used by themselves, they are intended to be included into other programs.

TOOLS

A list of the tools provided is as follows:

- A Python library (pysbml). This library provides a console based modelling system in Python, a tool for the visualization

of an SBML model (Fig. 1), and a tool for converting an SBML model to an HTML file. More information on both installing and using pysbml can be found in the documentation, available on-line at the authors' website.

- A stochastic simulator written in ANSI C (gillespie2). The algorithm executes the standard Gillespie algorithm (Gillespie, 1977). A swig interface has also been provided to allow the simulator to be imported into Python.
- SBML-shorthand provides a shorthand notation for SBML that is much easier for humans to read and write than full SBML. The full specification for SBML-shorthand and a conversion tool is available at the authors' website.
- Additionally, a variety of tools have been exposed as web-services, these include model visualization (Fig. 1), validation and conversion to an HTML document for display within a web browser. The WSDL file for these services can be found at <http://www.basis.ncl.ac.uk/sbml.wsdl>

The library pysbml is written in Python. Python is a dynamically typed, object-oriented, interpreted programming language useful for a broad range of tasks, ranging from file parsing to rapid development of large applications. Compared with most programming languages, Python is easy to learn and is becoming the language of choice for many novice programmers (who quickly discover the power of Python for scientific computing). Owing to this, Python is an ideal choice as a 'glue' language in large projects. The library can easily be installed in the standard manner and provides access to the simulator, visualization function and model builder.

The stochastic simulator (gillespie2) is built using the efficient GNU scientific libraries and libSBML. It currently supports local and global parameters, events (without delays) and assignment rules. Although the simulator can be called from the command line, it is envisaged that the simulator will form part of larger tool, and has a programmers API. For example, the simulator is being used as a component of the Biology of Ageing e-Science Integration and Simulation system currently being developed here at Newcastle [Kirkwood *et al.* (2003), and see Proctor *et al.* (2005) for a biological application of this system].

The library also includes the SBML-shorthand to SBML and SBML to SBML-shorthand Python conversion tools. These are useful for rapidly building and editing models destined for SBML encoding. They are particularly well-suited to building SBML models designed for discrete stochastic simulation.

*To whom correspondence should be addressed.

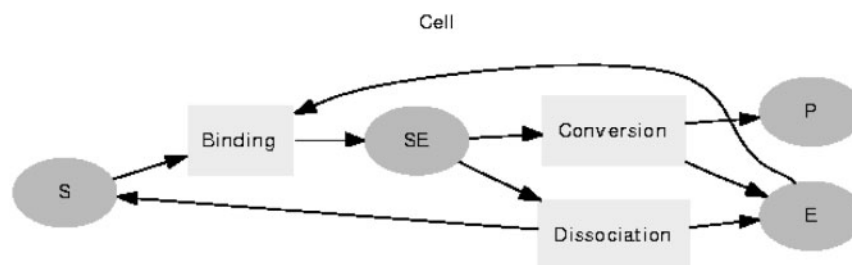


Fig. 1. Diagram of a simple biochemical network produced by the visualization tool.

We have exposed several methods as web-services. This includes an SBML validation tool, a visualization tool and an SBML to HTML converter. Using web-services, these tools can be accessed in a language and platform independent manner. For example, although the services were built using Python on Linux, they can be called using Perl or Java on Windows. Example scripts of how to call the services are given at the authors' website.

APPLICATION AREAS

We do not envisage that all the tools presented in this application note would be incorporated into any one SBML application. Rather, we foresee users 'picking and choosing' the tools which they require.

For example, pysbml compliments the PySCeS toolkit (Olivier *et al.*, 2005). PySCeS, also written in Python, focuses mainly on deterministic and steady-state behaviour, with additional support for metabolic control analysis. Since both tools are written in Python, combining them is straightforward. Other attractive Python toolkits are ScrumPy (<http://mudshark.brookes.ac.uk/ScrumPy>) and SloppyCell (<http://sloppycell.sourceforge.net>). Furthermore, pysbml can be easily incorporated into the Systems biology workbench (SBW) via the SBW Python API (Sauro *et al.*, 2003).

Another tool that is freely available is ODESolver. This provides a very fast and efficient library for solving continuous time ODEs and is a natural companion to the stochastic simulator, gillespie2.

Since the web-services can be accessed by most platforms and languages, possible usage is wide and varied. A simple example would be to use standard PHP modules to dynamically validate and visualize SBML models.

SUMMARY

We believe that our tools provide a valuable addition to the suite of applications that are becoming available for the exploration and understanding of biochemical networks. By providing multiple tools in smaller packages, rather than one large package, we enable users to 'pick-and-choose' according to their needs and usage. Furthermore, it is becoming possible to combine multiple libraries for modelling using SBML with little effort. However, there is still a need for further development and research.

ACKNOWLEDGEMENTS

We thank the BBSRC, MRC, DTI and Unilever for financial support and Jo Mathews for discussions relating to visualization software. We also thank the anonymous referees for their helpful comments for improving this note.

Conflict of Interest: none declared.

REFERENCES

- Gillespie, D.T. (1977) Exact stochastic simulation of coupled chemical-reactions. *J. Phys. Chem.*, **81**, 2340–2361.
- 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.
- Kirkwood, T.B.L. *et al.* (2003) Towards an e-biology of ageing: integrating theory and data. *Nat. Rev. Mol. Cell Biol.*, **4**, 243–249.
- Olivier, B.G. *et al.* (2005) Modelling cellular systems with PySCeS. *Bioinformatics*, **1**, 560–561.
- Proctor, C.J. *et al.* (2005) Modelling the actions of chaperones and their role in ageing. *Mech. Ageing Dev.*, **126**, 119–131.
- Sauro, H.M. *et al.* (2003) Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration. *OMICS*, **7**, 355–372.