Systems biology

Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology

Henning Schmidt* and Mats Jirstrand

Fraunhofer-Chalmers Research Centre for Industrial Mathematics, Chalmers Science Park,
SE-41288 Göteborg, Sweden

ABSTRACT

Summary: We present a Systems Biology Toolbox for the widely used general purpose mathematical software MATLAB. The toolbox offers systems biologists an open and extensible environment, in which to explore ideas, prototype and share new algorithms, and build applications for the analysis and simulation of biological and biochemical systems. Additionally it is well suited for educational purposes. The toolbox supports the Systems Biology Markup Language (SBML) by providing an interface for import and export of SBML models. In this way the toolbox connects nicely to other SBML-enabled modelling packages. Models are represented in an internal model format and can be described either by entering ordinary differential equations or, more intuitively, by entering biochemical reaction equations. The toolbox contains a large number of analysis methods, such as deterministic and stochastic simulation, parameter estimation, network identification, parameter sensitivity analysis and bifurcation analysis.

Availability: The Systems Biology Toolbox for MATLAB is open source and freely available from http://www.sbtoolbox.org. The website also contains a tutorial, extensive documentation and examples.

Contact: henning@fcc.chalmers.se

1 INTRODUCTION

Modelling efforts within Systems Biology lead to models containing an increasing number of components and associated interactions. The results are very complex, non-linear mathematical models of biological processes and systems, where the increasing complexity requires software that can provide powerful numerical analysis methods to investigate the behaviour of these models. Another important requirement for software tools, focusing on the mathematical analysis of biochemical systems, is flexibility. By this we, here, suggest that the target user (the user who is supposed to use the tool in question) should not be limited excessively by the tool. Instead, the user should be able to easily extend the functionality of the tool according to the users' needs by, for example, adding new functionality or by customizing existing features.

Current system biological analysis software often consists of relatively inflexible software applications that are either not extensible by the user and/or require considerable operating system-dependent programming knowledge in order for the user to be able to implement new methods. This kind of specialized programming

knowledge cannot be expected from the target user of a tool. Unless a certain tool, or a suite of tools, does contain all the functionality a target user is expecting, the above mentioned leads to a limitation for the user. Prototyping and implementation of new analysis methods become difficult and eventually the user has to wait until the developers of a certain tool have included a new method.

The Systems Biology Toolbox has been developed as a toolbox for MATLAB and aims at being a user-friendly and user-extensible, software-based, mathematical analysis framework for biological and biochemical systems. The toolbox enables the user to access all data and data structures, resulting in full control over the tasks to be performed and the possibility of focusing on the tasks one is interested in, leading to a faster workflow and accelerated scientific advancement.

MATLAB (http://www.mathworks.com) is a *de facto* standard in many scientific areas and already widely used in Systems Biology. It provides numerous state-of-the-art mathematical and numerical methods and a user-programmable platform, using a simple, but powerful, high-level scripting language that avoids programming overhead usually present when developing standalone software applications. Users will find it easy to use the functionality of the Systems Biology Toolbox. In order for a user to extend the functionality of the toolbox, knowledge about the MATLAB scripting language is required. However, users who are not familiar with MATLAB in advance will find it much easier to learn this scripting language than to learn how to program operating system-dependent standalone applications.

2 FEATURES

The Systems Biology Toolbox is built in a modular way, as depicted in Figure 1. The base elements are objects of classes SBmodel and SBdata, which are used to represent models and experimental data. The internal representation of SBmodels is based on ordinary differential equations (ODEs), limiting the type of models that can be represented to non-spatial ones. The model elements are parameters, variables, reactions, functions, discrete state events and ODEs. Additionally SBmodels may contain MATLAB functions that can be used by the toolbox to implement certain analysis methods. The latter feature leads to a high degree of flexibility of an SBmodel. Low-level functions are implemented as methods of the two object classes and include functions for importing and exporting models and data, along with the creation of these objects. An SBmodel can be imported from and exported to two different textual descriptions

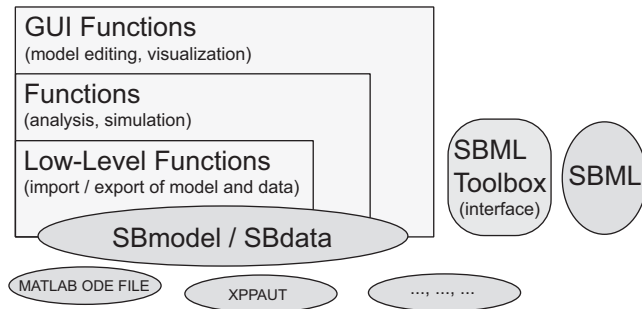*To whom correspondence should be addressed.

Fig. 1. Modular structure of the toolbox. The toolbox is designed in a modular way. Optionally, freely available third party software packages are used, e.g. to (i) perform bifurcation analysis and (ii) realize an interface to SBML.

of the same model. One representation is based on biochemical reaction equations, well suited for biochemists and biologists. The other representation is based on differential equations. Both representations are interchangeable. The toolbox can deal with Systems Biology Markup Language (SBML) models (http://sbml.org/), strongly facilitating the collaboration and the model exchange between researchers within the systems biology community. Import and export of SBML models is realized via the freely available SBML Toolbox (Keating, http://sbml.org/software/sbmltoolbox/), which is used to convert an SBML model to a MATLAB structure and vice versa. SBdata objects can be imported from and exported to Excel spreadsheets and comma separated value (CSV) data files.

Owing to the structured internal representation of SBmodels it is straightforward for users to write their own export functions in order to export an SBmodel into additional formats, based on ordinary differential equations. An example is the export to Maple (http://www.maplesoft.com) or C/C++. The advantage is that the same model does not have to be coded manually in different formats, but that only one original model needs to be present and can be exported to desired formats.

In the toolbox we have tried to limit the use of graphical user interfaces (GUIs). The main reason for this is that Mathworks often changes the underlying format for GUIs in new major releases, and that by limiting the complexity of the GUIs we can more easily handle upgrades to new MATLAB versions. So far there are MATLAB GUIs for visualization and editing of models, displaying simulation results and visualization of experimental data and analysis results. The first JAVA-based GUI was developed for the SBML export function.

The toolbox functions can be broken down into three different classes of functions: auxiliary functions, information and editing functions, and analysis functions.

Auxiliary functions are functions that support the other functions. For example, a non-linear solver, based on Newton iterations, local and global optimization functions, based on Nelder–Mead downhill simplex and simulated annealing (Press *et al.*, 1992), but also functions that help with the handling of files, and the interpretation of certain SBML elements.

Information functions are functions that extract information from an SBmodel or an SBdata object. Examples are functions that return state or reaction names along with corresponding right-hand sides of the ODEs or reaction rate expressions. Several functions are also available to edit models, for example, by setting new initial conditions or by changing parameter values.

The analysis functions include functions for deterministic and stochastic simulation and a more *in silico* experiment-oriented type of simulation. Stochastic simulation is available only on Windows platforms and is implemented by using a standalone executable that can be found at http://dion.che.udel.edu/multiscale/software.html. There are functions for the calculation of steady states, moiety conservations, Jacobians and stoichiometric matrices. More complex analysis functions include network identification (Schmidt *et al.*, 2005), and a suite of functions for parameter estimation. The user can use the default functionality, but can also define own cost-functions and use custom optimization methods. Parameter sensitivities can be computed for oscillating systems in terms of amplitude and period sensitivities, and in addition parametric steady state sensitivities can be determined (Varma *et al.*, 1999). Note that within the area of computational biology parameter sensitivity analysis is often called metabolic control analysis. Other analysis functions allow the localization of feedback mechanisms that are the source of complex behaviours, such as limit cycles and the occurrence of multiple steady states (Schmidt and Jacobsen, 2004). The toolbox also provides a function for performing bifurcation analysis via the freely available third party software XPPAUT (Ermentrout, http://www.math.pitt.edu/bard/xpp/xpp.html).

3 IMPLEMENTATION

The Systems Biology Toolbox requires at least MATLAB Release 14. No compilation is required and the use of MATLAB as a basis allows the toolbox to be used on Windows, Linux, Unix and MAC OS machines. The installation of the toolbox consists of unpacking a compressed file to the desired location and to edit path information in two scripts. Extensive documentation, examples and a tutorial are available on the toolbox' website (http://www.sbtoolbox.org). The toolbox is distributed under the GNU General Public Licence (Free Software Foundation 1991; http://www.gnu.org). The Systems Biology Toolbox does not require the presence of any other MATLAB toolbox.

For the import/export of SBML models, bifurcation analysis and stochastic simulation, the corresponding third party software tools need to be additionally installed.

ACKNOWLEDGEMENTS

REFERENCES

Ermentrout,G.B. (2000) *XPPAUT*. University of Pittsburgh, USA.

Keating,S. (2004) *SBML Toolbox*. University of Hertfordshire, UK.

Press,W.H., Teukolsky,S.A., Vetterling,W.T. and Flannery,B.P. (1992) *Numerical Recipes in C, The Art of Scientific Computing*. 2nd edn. Cambridge University Press, Cambridge.

Schmidt,H. and Jacobsen,E.W. (2004) Linear systems approach to analysis of complex dynamic behaviours in biochemical networks. *IEE Syst. Biol.*, **1**, 149–158.

Schmidt,H. *et al.* (2005) Identification of small scale biochemical networks based on general type system perturbations. *FEBS J.*, **272**, 2141–2151.

Varma,A., Morbidelli,M. and Wu,H. (1999) *Parametric Sensitivity in Chemical Systems*. Cambridge University Press, Cambridge.