SBMLToolbox: an SBML toolbox for MATLAB users
Sarah M. Keating¹,*, Benjamin J. Bornstein², Andrew Finney³ and Michael Hucka⁴
¹Science and Technology Research Institute, University of Hertfordshire, Hatfield, UK, ²NASA Jet Propulsion Laboratory, Pasadena, CA, USA, ³Physiomics PLC, Oxford, UK and ⁴Biological Network Modeling Centre, Beckman Institute 139-74, California Institute of Technology, Pasadena, CA, USA
Received on February 14, 2006; revised on March 20, 2006; accepted on March 21, 2006
Advance Access publication March 30, 2006
Associate Editor: Golan Yona

ABSTRACT
Summary: We present SBMLToolbox, a toolbox that facilitates importing and exporting models represented in the Systems Biology Markup Language (SBML) in and out of the MATLAB environment and provides functionality that enables an experienced user of either SBML or MATLAB to combine the computing power of MATLAB with the portability and exchangeability of an SBML model. SBMLToolbox supports all levels and versions of SBML.
Availability: SBMLToolbox is freely available from http://sbml.org/software/sbmltoolbox
Contact: s.m.keating@herts.ac.uk

1 INTRODUCTION
The expanding field of Systems Biology has stimulated the formalization of an increasing number of biological/biochemical models. The Systems Biology Markup Language (SBML), an XML-based format for computational models of biochemical networks, is becoming accepted as a de facto standard for the representation of such models (Hucka et al., 2004) and thus facilitates their systematic exchange.

In addition to promoting the creation of models, Systems Biology has also motivated the development of a range of software packages that can interact with these models, perform simulations and analyses on them, produce graphical representations of models and facilitate the creation of new models. However, the commercially available software package MATLAB (http://www.mathworks.com) provides a wide spectrum of this type of functionality combined with the facility to easily develop user-specific functions. Thus an alternative approach to that of developing new software exists in the form of developing a toolbox that provides users with an interface between basic MATLAB data structures and a format such as SBML. This not only enables users to leverage their existing skills in using the environment to work with a new format such as SBML, but also provides a substrate enabling other analysis tools in the environment to be applied to data represented in SBML.

MATLAB is a particularly attractive environment in this regard because it is already popular worldwide in both engineering and scientific research, and as the field of Systems Biology continues to attract researchers with an engineering or physical science background, the use of MATLAB within the field is likely to proliferate. Thus an alternative approach to that of developing new software is to develop a toolbox that provides users with an interface between basic MATLAB data structures and a format such as SBML. This not only enables users to leverage their existing skills in using the environment to work with a new format such as SBML, but also provides a substrate enabling other analysis tools in the environment to be applied to data represented in SBML.

2 FEATURES
The core functions of SBMLToolbox are for importing and exporting SBML models. The toolbox also contains a range of functions that allow a user to manipulate the content of the resulting MATLAB data structures (that is, the model being considered). A set of validation functions allow the structure to be checked for appropriate fields and the values assigned to these fields.

2.1 Import and export
SBMLToolbox offers two import functions that take an SBML model and return a MATLAB structure that replicates the Unified Modelling Language (UML) definition of an SBML model. Although the structure returned by both import functions is identical, two versions of this function are included within the toolbox: one in which the SBML file is verified against the XML Schema (Bray et al., 2000, http://www.w3.org/TR/1998/REC-xml-19980210) for SBML (see http://sbml.org/xml-schemas) and a number of consistency checks are performed prior to import, and a second version in which the verification stage is skipped.

In the toolbox, the returned structure is called a MATLAB_SBML structure. It is populated with values drawn from the model being imported. Each of the SBML ‘listOf’ elements is reproduced as an

*To whom correspondence should be addressed.
array of structures defining the particular element, whereas each
element is itself another structure. Figure 1 illustrates a portion of
MATLAB_SBML corresponding to a reaction and its relationship to
the definition of SBML’s Reaction.

For export, SBMLToolbox provides a function for translating a
MATLAB_SBML structure back to an XML file in SBML format.
As a convenience to users, the toolbox also provides additional
functions allowing a MATLAB_SBML model structure to be written
as a MATLAB data file known as a MAT file. This allows faster
access to stored models. The use of a MATLAB MAT file also
facilitates using MATLAB external library functions for the
manipulation of these data files. SBMLToolbox includes functions
for browsing models stored in MAT files.

2.2 Manipulation functions

One set of functions act directly on the structures and mimic the
functions provided by libSBML, the library developed and main-
tained by the SBML Team as part of the software infrastructure
supporting SBML development (Bornstein et al., 2003, http://sbml.
org/software/libsbml). Many developers of SBML compliant soft-
ware are familiar with libSBML and thus this set of functions gives
them an easy transition into the MATLAB environment. The func-
tions include those for creating objects, e.g. Model_create; func-
tions for setting field values, e.g. Compartment_setId; functions for
returning field values, e.g. Reaction_getId; functions for querying
whether a field value has been set, e.g. Species_isSetId and func-
tions for adding structures to arrays within the given structure,
e.g. Model_addRule.

A second set of manipulation functions in SBMLToolbox drive a
set of graphical user interfaces (GUI) that display the field names
and corresponding values for each element. For example, the
ViewSpecies function produces a GUI that shows the name/id of
the species, the compartment containing it, the initial concentration
and units and so on. The GUI does not provide editing functionality;
it is intended principally as a simple tool for displaying model
structures, and as a starting point for users’ development of more
elaborate capabilities.

In order to interact with the MATLAB_SBML model structure it
is necessary to extract the required information. The use of the
standard MATLAB structure allows easy access to field names
and indexing into arrays of structures within structures, e.g.
the name of the second species in ‘model’ can be accessed as
model.species(2).name. However the interaction with a model
may be more complex and SBMLToolbox provides a third set
of designed functions to return more detailed information from
the model; e.g. GetRateLawsFromReactions returns two arrays,
one of the names of each species in the model and the other con-
taining the character representation of the rate law used to determine
each species in the same order as the species, thus allowing the two
to be paired. These functions are also replicated to return symbolic
representations of the model information which facilitates the use
of MATLAB’s Symbolic Toolbox—an additional commercial toolbox
from The Mathworks that incorporates symbolic computation into
the MATLAB environment.

Finally, the MATLAB_SBML structure stores any formula/
MathML in the original model as a string representation of
the formula. In order to support interaction with the model,
SBMLToolbox contains a set of ‘convenience’ functions that facil-
itate the manipulation of the string representations of formula; e.g.
PairBrackets that determines the order in which brackets within a
formula should be considered and SubstituteFunction which allows
arguments to be passed to a formula defined by a function definition.

2.3 Simulation

SBMLToolbox is not intended to be a simulation tool; however, it
does contain a set of example functions that allow the user to
simulate an SBML model using MATLAB’s built-in ODE solvers.
The functionality provided includes writing out a script for use with
the ODE solvers, running the solvers, and displaying the data, as
well as a number of supporting functions. These functions deal with
all aspects of an SBML model. The simulation output passes all the
tests in the test-suite produced by the SBML Team (Finney et al.,
tion is that the package does not currently handle the ‘delay’ feature
in SBML.
3 IMPLEMENTATION

SBMLToolbox works with all versions of MATLAB from version 6.5 onwards and depends on the Xerces-C version of libSBML. The import/export functions do require compilation; however, scripts are included within the toolbox to facilitate this. There are two downloads available: a Windows installer that pre-packages the necessary libraries and precompiled functions, and a ‘source’ download that can be used to install the toolbox on Linux, Unix or Mac OS platforms. The functions providing interaction between a model and the symbolic toolbox obviously require the presence of the MATLAB Symbolic Toolbox; however, these functions are self-contained and the full functionality of the toolbox can be achieved without using symbols. In addition, the source download contains a number of test files, thus facilitating further development of the toolbox functions. Full documentation is included with SBMLToolbox and each function has extensive MATLAB help.

SBMLToolbox is distributed under the GNU Lesser General Public License (http://www.gnu.org/licenses/licenses.html#LGPL).

ACKNOWLEDGEMENTS

This work was supported by the BBSRC e-Science Initiative UK and grant number R01 GM70923 from the NIH National Institute of General Medical Sciences USA.

Conflict of Interest: none declared.

REFERENCES

Bornstein, B et al. (2003) LibSBML, a library designed to help you read, write, manipulate, translate and validate SBML.
Prajna, S et al. (2004) SOSTOOLS: sum of squares optimization toolbox for MATLAB.