

SBML Export Interface for the Systems Biology Toolbox for MATLAB

Henning Schmidt¹, Gunnar Drews², Julio Vera², Olaf Wolkenhauer²

¹Fraunhofer-Chalmers Research Centre for Industrial Mathematics, Gothenburg, Sweden

²Systems Biology and Bioinformatics Group, Department of Computer Science, University of Rostock, Rostock, Germany

ABSTRACT

Summary: In this application note we present an SBML export interface for the Systems Biology Toolbox for MATLAB. This interface allows modelers to automatically convert models, represented in the toolbox's own format (SBmodels) to SBML files. Since SBmodels do not explicitly contain all the information that is required to generate SBML, the necessary information is gathered by parsing SBmodels. The export can be done in two different ways. Firstly, it is possible to call the export from the command line, thereby directly converting a model to an SBML file. The second option is to inspect and edit the conversion results with the help of a graphical user interface, and to subsequently export the model to SBML.

Availability: The SBML export interface has been integrated into the Systems Biology Toolbox for MATLAB, which 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

The Systems Biology Markup Language (SBML) (1) is a computer-readable language for the representation of mathematical models of biochemical networks, which in recent years has become a de-facto standard for the description and exchange of mathematical models in systems biology. The aim of SBML is to facilitate the exchange of mathematical models between software tools with a minimum of human intervention and to allow an easy publication in an electronic format.

The Systems Biology Toolbox for MATLAB (2) is an environment designed for modelling, simulation, and computational analysis of biochemical systems. The central part of the toolbox is the basic data structure that is used to represent the mathematical models (SBmodel).

SBML defines a model in terms of its components, such as species, parameters, and compartments, and the interactions between these components, such as reactions. In contrast, an SBmodel represents a model in terms of its underlying mathematical equations. Elements in an SBmodel are state variables, differential equations, constant parameters, and algebraic variables. While the import of SBML models into the SBmodel format is straightforward, the export is complicated by the fact that an SBmodel lacks explicit information that is necessary to do the mapping. For example, an SBML species can correspond mathematically to a state variable, a variable defined by an algebraic expression, or a constant parameter, depending on the definitions in the SBML file. This means

that in order to be able to export an SBmodel to SBML, additional information needs to be added to the SBmodel.

Subfield	Valid content	Condition
type	'isSpecie'	if SBML species
	'isParameter'	if SBML parameter
	'isCompartment'	if SBML compartment
compartment	species compartment	if type='isSpecie': name of the compartment where the SBML species is located in
	outside compartment	if type='isCompartment': name of the outside compartment. Leave empty if no outside compartment
	"	if type='isParameter'
unittype	'amount'	if species given in amount units, e.g., mmol
	'concentration'	if species given in amount units, e.g., mM
	"	if type is not 'isSpecie'

Table 1. Overview of additional information that is required for an SBmodel in order to export it to an SBML file.

More specifically, this additional information needs to be provided for all state variables, variables, and parameters of the SBmodel. In the data structure that is underlying an SBmodel the `states`, `variables`, and `parameters` fields each have the subfields `type`, `compartment`, and `unittype`, which need to be filled with valid content, defined in Table 1.

For the user there are two ways to manually add this information to the SBmodel format. The first possibility is to directly access the data structure of the SBmodel and to add the information there. The second, more convenient, method is to specify this information directly in the text description of an SBmodel. Figure 1 shows a simple example, where the additional information is highlighted in bold face.

We need to point out that, apart from the discussed elements of a model, other aspects, such as events and functions, are present, both in the SBML and SBmodel description. In this paper, however, we limit ourselves to the discussion of the most important aspects, since the consideration of the other elements during export is trivial.

2 SBML EXPORT FUNCTIONS

The SBML export interface, presented in this note, enables the user to automatically convert SBmodels to SBML files, without the

```

***** MODEL STATES
d/dt(S1) = -R [isSpecie:cytosol:concentration]
d/dt(S2) = R [isSpecie:nucleus:concentration]
d/dt(X) = -0.5*X + Y [isParameter]

***** MODEL PARAMETERS
k1 = 1 [isParameter]
k2 = 2 [isParameter]
cytosol = 10 [isCompartment:]
nucleus = 1 [isCompartment:cytosol]

***** MODEL VARIABLES
Y = 1-S1 [isParameter]

***** MODEL REACTIONS
R = k1*S1 - k2*S2

```

Fig. 1. Textual description of a sample SBmodel, showing the additional information, required for SBML export, in bold face.

need for manually adding the additional information. However, as human creativity is nearly unlimited, manual intervention is sometimes necessary. In the export interface this is achieved in two ways. Firstly, the user can add parts of the additional information into the model, as described above. This initial information will not be overwritten by the parsing step of the export function. A graphical user interface (GUI), shown in Figure 2, has been developed to inspect and edit all details of the model together with the parsed additional information.

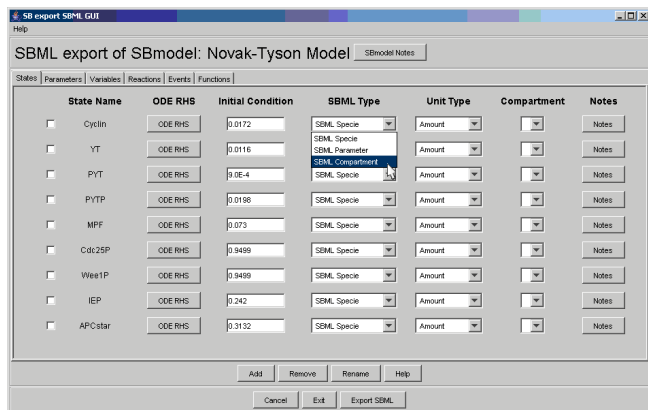


Fig. 2. Graphical user interface of the SBML export module.

The direct export of an SBmodel to an SBML file is performed by using the command:

```
» SBexportSBML(model)
```

The GUI based export functionality is invoked by calling:

```
» SBexportSBMLGUI(model)
```

where the variable `model` represents a valid SBmodel.

3 LIMITATIONS AND HINTS

The main limitations of the SBML export interface stem from limitations in the Systems Biology Toolbox. The export is done only to SBML level 2 version 1; unit definitions are not taken into account,

state events can not have a time delay assigned to them. Furthermore, algebraic rules and non-numeric stoichiometric coefficients are not supported. SBmodels allow the user to use pure MATLAB code in the *MATLAB functions* section of an SBmodel. Since these functions cannot be represented in SBML, the export of such models stops with a warning. Additionally, in the mathematical description of SBmodels the call of MATLAB functions is allowed. However, not all of these functions might have MathML equivalents.

In order to be able to correctly detect the type of a state in an SBmodel, to parse the stoichiometric coefficients, to detect the presence of eventual compartments in the model, and to decide if a certain species state-variable has *amount* or *concentration* units the differential equations need to be specified in a certain syntax, explained below.

$$d/dt(A) = R1 + 2*R2$$

Under the assumption that $R1$ and $R2$ are defined as reactions in the SBmodel, A is parsed as a specie. Since reaction rates in SBML are defined to have *amount/time* units, A must have *amount* units. The stoichiometric coefficients of the reactions are given by the numeric factors of the reactions.

$$d/dt(A) = (R1 + 2*R2)/C$$

Using this syntax, again under the assumption that $R1$ and $R2$ are defined as reactions, C is interpreted as a compartment size and the unit-type of A is set to concentration. In the case that in the right-hand-side terms appear that do not fit into this syntax ($R1$ could for example be defined as a parameter or a variable), A would be determined to be an SBML parameter and the differential equation would be implemented as a rate rule in SBML.

Finally, the hierarchy of compartments cannot be determined automatically. This means, if compartments are nested within a model, this information has to be added manually, either prior to calling the export function or by using the GUI. An SBmodel does not require the presence of a compartment. In the case that no compartment is present the export functions introduce a compartment with the name `default`.

4 IMPLEMENTATION

The export interface has been integrated into the Systems Biology Toolbox for MATLAB, which requires at least MATLAB Release 14. Furthermore, the SBML Toolbox (3) needs to be installed, since the developed SBML export functions convert an SBmodel into a MATLAB structure and subsequently use the `OutputSBML` command from the SBML Toolbox to write out the SBML file. The graphical user interface is implemented in JAVA. The Mathworks SimBiology Toolbox is not required.

REFERENCES

- [1]M. Hucka, *et al.*: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models, *Bioinformatics*, 2003, **19**(4), pp. 524-531.
- [2]Schmidt, H., Jirstrand, M. (2006) Systems Biology Toolbox for MATLAB: A computational platform for research in Systems Biology, *Bioinformatics*, Application Note, **22**(4), 514-515, <http://www.sbtoolbox.org>.
- [3]Keating, S. *SBML Toolbox*, University of Hertfordshire, UK, <http://sbml.org/software/sbmltoolbox/>.