

## MOCCASIN: converting MATLAB ODE models to SBML

Michael Hucka<sup>1,\*</sup>, Harold F. Gómez<sup>2,\*</sup>, Sarah M. Keating<sup>3,\*</sup>, Dagmar Iber<sup>2</sup>, Stuart Sealfon<sup>4</sup>

<sup>1</sup>Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA,

<sup>2</sup>Department of Biosystems Science and Engineering, ETH Zürich, Basel, 4058, CH, <sup>3</sup>European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK, <sup>4</sup>Department of Neurology, Mount Sinai School of Medicine, New York, NY 10029, USA

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

---

### ABSTRACT

**Summary:** MATLAB is a prevalent and well-supported numerical computing environment commonly used to implement and simulate ordinary differential equation (ODE) models. Despite a number of software efforts, translating MATLAB ODE models into open standard formats in system biology, such as the Systems Biology Markup Language (SBML), still remains a challenging task for modelers who have legacy models and need a straightforward solution. We introduce MOCCASIN (Model ODE Converter for Creating Awesome SBML Interoperability) - an open-source tool that uses a combination of heuristics and user assistance to convert ODE models written in MATLAB into SBML format.

**Availability and Implementation:** MOCCASIN is released under the terms of the LGPL 2.1 license (<http://www.gnu.org/licenses/lgpl-2.1.html>). Source code, binaries and test cases can be freely obtained at <https://github.com/sbmlteam/moccasin>.

**Supplementary Information:** More information regarding the general implementation of MOCCASIN can be found in the user guide at <https://github.com/sbmlteam/moccasin/tree/master>.

**Contact:** [moccasin-dev@googlegroups.com](mailto:moccasin-dev@googlegroups.com)

### 1 INTRODUCTION

The *Systems Biology Markup Language* (SBML) is a widely accepted open-source format for representing and exchanging data, diagrams and models across analysis and simulation tools in systems biology (Hucka *et al.*, 2003). Born from the need to resolve incompatibilities between software platforms that use different formats to describe model components, SBML is neutral with respect to modeling framework and computational platform – which helps ensure portability and integration across tools. This has led to over 250 different software tools adopting SBML as a file format for model encoding.

Given the growing body of ordinary differential equation (ODE) models being implemented in MATLAB (Ghosh, S. *et al.*, 2011) as well as the benefits of SBML model description (Hucka *et al.*, 2003), several software efforts have tried to close the gap that prevents direct translation of ODE MATLAB models into SBML. Popularly used tools include MATLAB toolboxes such as the Systems Biology Toolbox (Schmidt and Mats, 2006) and the SBML Toolbox (Keating *et al.*, 2006). While these are high-quality resources that offer a given degree of indirect SBML manipulation from MATLAB, they are limited as they either lack support for handling the latest SBML (L3V1), or are difficult to use when researchers simply want to convert already-existing MATLAB scripts into SBML models. As a result, there is a need for a tool that allows for straightforward translation of existing MATLAB ODE models into SBML, without requiring adherence to strict MATLAB syntax or code modifications for translating complex models.

In this article, we introduce MOCCASIN (the Model ODE Converter for Creating Awesome SBML Interoperability) – a stand-alone tool that intuitively helps researchers take ODE models written in MATLAB and Octave, and export them as SBML files. With a highly modularized design, MOCCASIN is able to parse MATLAB code- without using MATLAB, and then export an equivalent SBML file by translating individual model constructs. A basic characteristic of this tool is that it does not intend to provide the ability to directly or indirectly manipulate MATLAB or SBML code, but rather serve as a translator that delivers SBML code with the same model elements and kinetics as a given ODE MATLAB file. Furthermore, MOCCASIN draws from recent theoretical advances to allow for the inference of biochemical reaction networks (Fages *et al.*, 2015). This allows for richer SBML that can also be used for qualitative analyses where knowledge of the reaction network behind a system of ODEs is needed.

## 2 IMPLEMENTATION

### 2.1 Modules

Written entirely in Python, MOCCASIN's design is entirely modular. The available pipeline is comprised of: (1) a module that parses input MATLAB files into an equivalent internal representation of the contained ODE models; (2) a module that reconstructs a complete model from parsed input and exports it either as an SBML model with rate equations for kinetics, or as an XPP file; and (3) a module that couples reconstructed input, a web service call to *The Biochemical Abstract Machine* (BIOCHAM) (Fages *et al.*, 2015), and a post-processing step in order to export SBML output with biochemical reactions for kinetics. Owing to this modularity, MOCCASIN is highly extensible, and flexible enough to allow developers to use as few or as many modules as they want. The aforementioned framework is illustrated by Figure 1, and described in detail below.

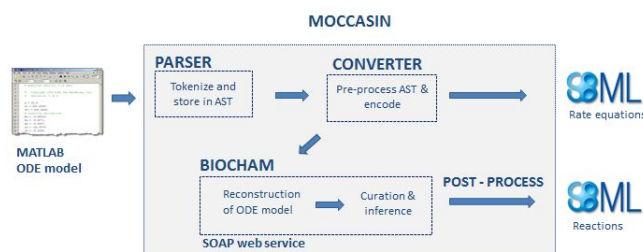


Figure 1: Schematic of constituent MOCCASIN modules.

#### 2.1.1 Parsing module

The first step in the MOCCASIN pipeline involves the parsing of input MATLAB code into its constituents. To this end, the grammar understood by the parser enables the seamless identification and annotation of constructs commonly found in MATLAB ODE models. Nevertheless, properly formatted input models are required to reside in a single file, set up a system of ordinary differential equations as a function defined in the same file, and make a call to one of the MATLAB odeNN family of solvers (e.g., ode45, ode15s, etc.). After being parsed, model components are hierarchically stored as nodes of an Abstract Syntax Tree (AST).

#### 2.1.2 Converter module

Following parsing and storage, the converter module proceeds to translate the internal AST-based representation of the input ODE model into significant output. There are two kinds of file formats that the converter is able to export: SBML and XPP. In the case of SBML, a model definition is assembled piece by piece from AST nodes using the libSBML library (Bornstein *et al.*, 2008). LibSBML is an *application programming interface* (API) library with Python bindings that not only allows for the import and export of SBML-based files, but for the manipulation of data structures patterned after SBML constructs - which further facilitates model building. Exported files retain the intended model semantics of the initial MATLAB ODE model, and use rate equations to describe their temporal evolution.

The converter module is also capable of exporting models into XPP file format (Ermentrout, 2002). XPP is used in the systems biology community to facilitate the numerically study of equation-based models. Outputted XPP model definitions are pieced together by converting AST nodes corresponding to relevant mathematical components into a set of declarations and equations that conform to XPP format rules.

#### 2.1.3 BIOCHAM module

Although encoding the ODE kinetics of an input MATLAB file using rate equations in SBML is sufficient to ensure simulation reproducibility, this overlooks the biochemical reaction network underlying said equation system. Furthermore, a variety of qualitative analyses, which do not rely on kinetic information but on the structure of the reactions, have rapidly developed for reasoning on large interaction networks. Among others, these include approaches to reduce models using graph theory concepts (Gay *et al.*, 2014) and reachability assessments in reaction networks (Fages *et al.*, 2004).

In order to provide export of SBML models with fully resolved reaction networks, this module integrates parsed input with BIOCHAM's web-service, and a post-processing step. BIOCHAM is a modeling environment for systems biology that encompasses the published implementation of the state-of-the-art algorithm for reconstructing and inferring the complete reaction model from a given set of ODEs (Fages *et al.*, 2015). Pipeline interaction with BIOCHAM is done via a Simple Object Access Protocol (SOAP) call, which enables the supply of parsed input in the form of an XPP file and the retrieval of output as L3V1 SBML code.

Due to syntactic limitations in XPP as well as BIOCHAM not accepting SBML with rate equations as input, retrieved SBML models do not contain several components present in the original model (i.e., time variables, initial assignments, units). In order to accurately translate the entire input ODE model and accommodate as wide a range of modeling scenarios as possible, this module post-processes collected BIOCHAM output before exporting to the user. In this way, all components of the initial MATLAB file are modeled and each reaction in the final SBML system is fully characterized with well-identified reactants, products and modifiers.

## 2.2 INTERFACES

MOCCASIN exposes a cross-platform, user-friendly command-line (CLI) and a graphical-user interface (GUI). Command-line runs may be customizable via flags which, for instance, allow the display of debugging information or enable the encoding of model variables as SBML parameters instead of SBML species. While the GUI is alone in providing interactive processing, users can use either interface to obtain XPP or SBML models that are characterized by either biochemical reactions or rate equations.

## 3 FUTURE WORK

Pipeline enhancements will focus on (1) expanding the grammar that the parser understands to enable the interpretation and encoding of MATLAB flow control constructs and comments; (2) allowing model pooling from several MATLAB input files; (3) generating SED-ML (The Simulation Experiment Description Markup Language) files, which are used to encode simulation parameters essential for ensuring reproducibility; and (4) re-implementing the state-of-the-art algorithm used by BIOCHAM to infer the biochemical reaction network underlying a set of input ODEs. .

## ACKNOWLEDGEMENTS

The authors would like to acknowledge Thomas B. Kepler, Franz-Josef Elmer, Bernd Rinn and Jannik Vollmer for their helpful discussions and implementation ideas.

*Funding:* This work was supported by the Modeling Immunity for Biodefense contract by the National Institutes of Health (NIH contract numbers HHSN266200500021C, awarded to Stuart Sealfon; and HHSN272201000053C, awarded to Thomas B. Kepler and Garnett H. Kelsoe). We also acknowledge support by the Swiss Institute of Bioinformatics.

*Conflict of Interest:* none declared.

## REFERENCES

- Bornstein, B.J. et al. (2008) LibSBML: an API Library for SBML. *Bioinformatics*, 24, 880-881.
- Ermentrout, B. (2002) Simulating, analyzing, and animating dynamical systems: a guide to XPPAUT for researchers and students. *Society for Industrial Mathematics*.
- Fages, F. et al. (2004) Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *J. Biol. Phys. Chem.*, 4 (2): 64-73.
- Fages, F. et al. (2015) Inferring Reaction Systems from Ordinary Differential Equations. *Theoretical Computer Science*, 22 (4): 514-515.
- Gay, S. et al. (2014) On the Subgraph Epimorphism Problem. *Discrete Applied Mathematics*, 162: 214-228.
- Ghosh, S. et al. (2011) Software for Systems Biology: From Tools to Integrated Platforms. *Nature Reviews Genetics*, 12: 821-832.
- 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.
- Keating, S.M. et al. (2006) SBMLToolbox: An SBML Toolbox for MATLAB Users. *Bioinformatics*, 22 (10): 1275-1277.
- Schmidt, H., and Mats J. (2006) Systems Biology Toolbox for MATLAB: A Computational Platform for Research in Systems Biology. *Bioinformatics*, 22 (4): 514-515.