Bioinformatics, 39(1), 2023, btad044
https://doi.org/10.1093/bioinformatics/btad044
Advance Access Publication Date: 20 January 2023
Applications Note

Systems biology

SBML2HYB: a Python interface for SBML compatible hybrid modeling

José Pinto [1,†], Rafael S. Costa [1,*,†], Leonardo Alexandre [1,2], João Ramos [1] and Rui Oliveira[1]

[1]LAQV-REQUIMTE, Department of Chemistry, NOVA School of Science and Technology, Universidade NOVA de Lisboa, Caparica 2829-516, Portugal and [2]INESC-ID, Lisboa, Portugal

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Janet Kelso

Received on August 2, 2022; revised on January 3, 2023; editorial decision on January 17, 2023; accepted on January 19, 2023

Abstract

Summary: Here, we present *sbml2hyb*, an easy-to-use standalone Python tool that facilitates the conversion of existing mechanistic models of biological systems in Systems Biology Markup Language (SBML) into hybrid semi-parametric models that combine mechanistic functions with machine learning (ML). The so-formed hybrid models can be trained and stored back in databases in SBML format. The tool supports a user-friendly export interface with an internal format validator. Two case studies illustrate the use of the *sbml2hyb* tool. Additionally, we describe HMOD, a new model format designed to support and facilitate hybrid models building. It aggregates the mechanistic model information with the ML information and follows as close as possible the SBML rules. We expect the *sbml2hyb* tool and HMOD to greatly facilitate the widespread usage of hybrid modeling techniques for biological systems analysis.

Availability and implementation: The Python interface, source code and the example models used for the case studies are accessible at: https://github.com/r-costa/sbml2hyb.

Contact: rs.costa@fct.unl.pt

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Hybrid semiparametric models combine parametric functions (stemming from knowledge) with non-parametric functions (stemming from data) in the same mathematical structure (Thompson and Kramer, 1994). The incorporated parametric functions have a fixed mathematical structure with a fixed number of parameters established from prior knowledge. On the contrary, the non-parametric functions have loose structure without physical meaning. They are frequently needed because critical parts of the model lack fundamental mechanistic knowledge. The number and value of such parameters are not known a priori. They must be established from data during the training process. A typical example is the combination of material balance equations over biochemical species in the form of a system of ordinary differential equations (ODEs)—parametric function easily established from prior knowledge—with an artificial neural network (ANN) to model partially or completely the biologic kinetics—non-parametric function of a more complex part of the model lacking mechanistic understanding (Pinto *et al.*, 2019, 2022).

Hybrid models are currently well established in process systems engineering (von Stosch *et al.*, 2014). The penetration of the hybrid modeling technique in systems biology is however lagging behind. We have previously published hybrid metabolic flux analysis techniques that combine metabolic networks with principal component analysis (Carinhas *et al.*, 2011; Isidro *et al.*, 2016) and partial least squares (Ferreira *et al.*, 2011; Teixeira *et al.*, 2011). The combination of systems of ODEs with ANNs for modeling biochemical networks with intrinsic time delays has been proposed by von Stosch *et al.* (2011). Recent studies have highlighted the need to further integrate systems biology models, particularly genome-scale models (GEMs), with emerging machine learning (ML) techniques (Antonakoudis *et al.*, 2020; Kim *et al.*, 2021; Lewis and Kemp, 2021; Ramos *et al.*, 2022; Vijayakumar *et al.*, 2020; Yang *et al.*, 2019).

A large number of systems biology models, including GEMs, have been developed and stored in databases [e.g. BioModels (Le Novere *et al.*, 2006) and JWS online (Olivier and Snoep, 2004)] in the Systems Biology Markup Language (SBML) format (Hucka *et al.*, 2003). However, existing hybrid modeling tools do not comply with the SBML format. This significantly hinders the interlink between both modeling approaches. Here, we develop the *sbml2hyb* export interface, which allows to convert existing systems biology models into a hybrid model and *vice versa*. Moreover, this work

presents a new internal hybrid model format (HMOD) that can be translated to SBML.

Figure 1 shows the pipeline for SBML-compatible hybrid modeling. The proposed workflow enables to convert existing systems biology models stored in databases in SBML format into hybrid models that combine mechanistic equations and ML techniques. SBML is not a common format to encode ML/hybrid models (i.e. multiple formalism models), thus we created an intermediate HMOD format (described below). The user inputs the information of the ML module into the HMOD format (currently limited to feedforward ANNs). The resulting hybrid model in HMOD format is reconverted in SBML and stored back in model databases. Hybrid models in SBML format can be simulated, analyzed, trained with existing tools such as MATLAB and COPASI (Hoops *et al.*, 2006) or special-purpose tools with training algorithms for hybrid systems which are able to read SBML files. To facilitate the conversion between the SBML and HMOD files, a Python-based interface is provided.

2 Methods and implementation

2.1 Overview of HMOD format

The HMOD format is a text-based file (ASCII) with the list of properties (species, parameters, rates and rules) defining the model that make it easy to parse. This format presents the model components in

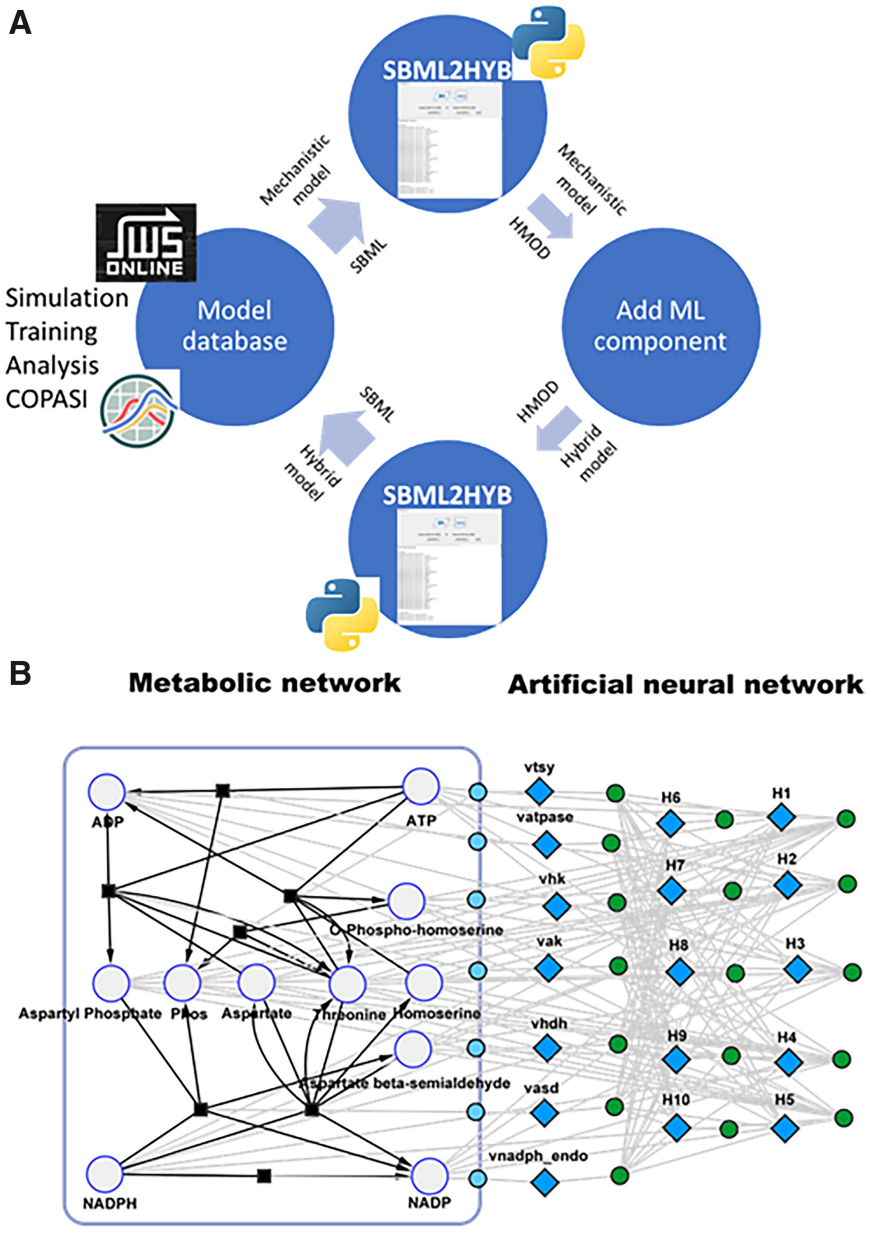

Fig. 1. SBML compatible hybrid modeling pipeline. (**A**) Overview of the *sbml2hyb* pipeline. Stored SBML (mechanistic) models in databases are converted to the HMOD format by the *sbml2hyb* tool. The user inputs the information of the ML component in the *sbml2hyb* interface (input/output variables and Keras neural network file in H5 format), which is then automatically added to the HMOD file. The resulting hybrid model HMOD file is reconverted to SBML. The hybrid model in SBML format is stored back in databases; (**B**) Simplified illustration of a hybrid model in SBML format generated by Cytoscape with the cy3sbml app (Konig *et al.*, 2012). On the mechanistic side of the model (left of the image), the larger circles represent the different species of the model, the black squares represent the reactions and the large rectangle refers to the single compartment in this case. On the machine learning side (right of the image), each of the small green circles is a calculation carried out by the ANN, while each blue diamond represents the *results* of those calculations (hidden and output layers). The small blue circles are the final output of the network, which, in this case, is the value that is assigned to each of the reaction rates

a similar manner to SBML, by considering any number of species with a certain initial concentration distributed among any number of compartments. These species are then subjected to a list of reactions and rate rules which in turn can be dependent on various parameters and assignment rules.

2.2 *SBML2HYB* tool

The *sbml2hyb* tool is implemented in Python 3, along with some open-source libraries for the translations from one file format to another. To parse the SBML files, libSBML (Bornstein *et al.*, 2008) and Python's built-in xml.etree submodule were used. The libSBML translates the MathML expressions to common readable math and *vice versa*. The xml.etree, is a powerful xml parser and builder, which was helpful to get the necessary information and the wanted nodes from the SBML in order to build the HMOD files. These two libraries were used to parse the SBML and get the information to build the HMOD files and also for the MathML translation and SBML file building.

To parse and validate the HMOD files, it was necessary to build a new syntax validator that ensured that the file followed all the rules defined by this file type. The libSBML was also used to translate simple math equations to the MathML used in the SBML files. Hence, two Python modules were constructed (i) a module that receives an SBML, validates the file, and builds the corresponding HMOD file and validates it and (ii) a module that receives an HMOD file and transforms it into an SBML, which is also validated. Any error occurring in the validation phase is displayed in the GUI. This can be because of some xml structure fault, or the HMOD file not respecting the rules defined by the HMOD specification. The GUI was built using the TKinter library (Shipman, 2010). Moreover, the executable Windows file was generated using cx_Freeze (https://cx-freeze.readthedocs.io/en/latest/). The *sbml2hyb* package can be easily extended with additional functionalities that can interoperate with those already implemented.

3 Case studies

In order to demonstrate the applicability of the *sbml2hyb* pipeline (Fig. 1), we used two case models taken from the literature. For the first case, the threonine synthesis pathway model of *Escherichia coli* (Chassagnole *et al.*, 2001) in SBML format is freely available at the BioModels Database (BIOMD0000000066) while for the second case [the well-known Park and Ramirez model (1990)] the SBML file was created.

Firstly, the SBML files were converted into HMOD files via the *sbml2hyb* tool. These files contained the necessary information from the SBML in an easy to work format (species, compartments, rules and reactions). Afterwards, the ML component information was added. With the HMOD format containing information from the SBML and ML, it was possible for the parameters and assignments related to ML to be written back into the HMOD file.

Lastly, the obtained HMOD file was translated back through the *sbml2hyb* tool to obtain SBML file with the implemented hybrid model (can be also trained). These SBML files were then uploaded back into the BioModels database (MODEL2207280001 and MODEL2211110001), with the results being very similar to the original mechanistic models (see Supplementary Figs S1 and S2), showing that the hybrid models were successfully trained and turned into an SBML format. All models are available from https://github.com/r-costa/sbml2hyb/tree/main/models.

4 Conclusion

Previously published hybrid modeling studies are limited to relatively simple mechanistic models in the hybrid mechanistic/ML ensemble and do not comply with the SBML format. In this paper, we presented *sbml2hyb*, a Python application for SBML-compatible hybrid model encoding. It is easy to use and allows creating further extensions that can easily incorporate new model components. We expect the *sbml2hyb* tool to greatly facilitate the extension of existing SBML mechanistic models to the hybrid mechanistic/ML approach. All in all, the possibility to encode hybrid models in SBML format will accelerate the adoption of the hybrid modeling techniques by the systems biology community.

Funding

This work was supported by the Associate Laboratory for Green Chemistry—LAQV which is financed by national funds from FCT/MCTES [UIDB/50006/2020 and UIDP/50006/2020]. This work has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement number [101000733] (PROMICON project). The authors thank H. Mochao for useful implementation ideas. JP and LA acknowledge PhD grants [SFRD/BD14610472019 and 2021.07759.BD], Fundação para a Ciência e Tecnologia (FCT) and RSC the contract [CEECIND/01399/2017].

Conflict of Interest: none declared.

References

Antonakoudis,A. *et al.* (2020) The era of big data: genome-scale modelling meets machine learning. *Comput. Struct. Biotechnol.*, **18**, 3287–3300.

Bornstein,B.J. *et al.* (2008) LibSBML: an API library for SBML. *Bioinformatics*, **24**, 880–881.

Carinhas,N. *et al.* (2011) Hybrid metabolic flux analysis: combining stoichiometric and statistical constraints to model the formation of complex recombinant products. *BMC Syst. Biol.*, **5**(34).

Chassagnole,C. *et al.* (2001) Control of the threonine-synthesis pathway in *Escherichia coli*: a theoretical and experimental approach. *Biochem. J.*, **356**, 433–444.

Ferreira,A.R. *et al.* (2011) Projection to latent pathways (PLP): a constrained projection to latent variables (PLS) method for elementary flux modes discrimination. *BMC Syst. Biol.*, **5**(181).

Hoops,S. *et al.* (2006) COPASI — a COmplex PAthway SImulator. *Bioinformatics*, **22**, 3067–3074.

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.

Isidro,I.A. *et al.* (2016) Hybrid metabolic flux analysis and recombinant protein prediction in *Pichia pastoris* X-33 cultures expressing a singlechain antibody fragment. *Bioprocess Biosyst. Eng.*, **39**, 1351–1363.

Kim,Y. *et al.* (2021) Machine learning applications in genome-scale metabolic modeling. *Curr. Opin. Syst. Biol.*, **25**, 42–49.

Konig,M. *et al.* (2012) CySBML: a cytoscape plugin for SBML. *Bioinformatics*, **28**, 2402–2403.

Le Novere,N. *et al.* (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, **34**, D689–D691.

Lewis,J.E. and Kemp,M.L. (2021) Integration of machine learning and genome-scale metabolic modeling identifies multi-omics biomarkers for radiation resistance. *Nat. Commun.*, **12**(1), 2700.

Olivier,B.G. and Snoep,J.L. (2004) Web-based kinetic modelling using JWS online. *Bioinformatics*, **20**, 2143–2144.

Park,S.J. and Ramirez,W.F. (1990) Effect of transcription promoters on the optimal production of secreted protein in Fed-Batch reactors. *Biotechnol. Prog.*, **6**, 311–318.

Pinto,J. *et al.* (2019) A bootstrap-aggregated hybrid semi-parametric modeling framework for bioprocess development. *Bioprocess Biosyst. Eng.*, **42**, 1853–1865.

Pinto,J. *et al.* (2022) A general deep hybrid model for bioreactor systems: combining first principles with deep neural networks. *Comput. Chem. Eng.*, **165**, 107952.

Ramos,J.R.C. *et al.* (2022) Genome-scale modeling of Chinese hamster ovary cells by hybrid semi-parametric flux balance analysis. *Bioprocess Biosyst. Eng.*, **45**, 1889–1904.

Shipman,J.W. (2010) *Tkinter 8.4 Reference: a GUI for Python*. New Mexico Tech Computer Center.

Teixeira,A.P. *et al.* (2011) Cell functional enviromics: unravelling the function of environmental factors. *BMC Syst. Biol.*, **5**(92).

Thompson,M.L. and Kramer,M.A. (1994) Modeling chemical processes using prior knowledge and neural networks. *AIChE J.*, **40**, 1328–1340.

Vijayakumar,S. *et al.* (2020) A hybrid flux balance analysis and machine learning pipeline elucidates metabolic adaptation in cyanobacteria. *Iscience*, **23**, 101818.

von Stosch,M. *et al.* (2011) A novel identification method for hybrid (N)PLS dynamical systems with application to bioprocesses. *Expert Syst. Appl.*, **38**, 10862–10874.

von Stosch,M. *et al.* (2014) Hybrid semi-parametric modeling in process systems engineering: past, present and future. *Comput. Chem. Eng.*, **60**, 86–101.

Yang,J.H. *et al.* (2019) A White-Box machine learning approach for revealing antibiotic mechanisms of action. *Cell*, **177**, 1649–1661.e9.