The Journal of Open Source Software

SMBLtoODEpy: A software program for converting SBML models into ODE models in Python

Steve M. Ruggiero[1] **and Ashlee N. Ford Versypt**[1]

1 School of Chemical Engineering, Oklahoma State University

DOI: 10.21105/joss.01643

Software

- Review
- Repository
- Archive

Submitted: 02 August 2019
Published: 20 September 2019

Summary

Systems Biology Markup Language (SBML) is a standard intermediate language for representing models of biological systems, particularly reaction networks (Finney & Hucka, 2003; Hucka et al., 2003). SBML defines models independent of the software used to create the model, which allows for easy import and export of models. Programs for converting models into code exist for a variety of software languages but have key limitations. For example, the Systems Biology Format Converter (SBFC) is a suite of tools used to convert SBML models into scripts in multiple modeling and programming languages (Rodriguez et al., 2016). The BioModels Database uses SBFC to automate conversion of a large library of SBML models into other formats that can be downloaded by users (Chelliah et al., 2015; Glont et al., 2018). SBFC does not include a tool for generating Python scripts. For the languages SBFC supports, the codes generated are stand alone implementations of models that are not easy to integrate into other projects.

The present work describes a software program called SBMLtoODEpy that we developed to address these limitations by enabling conversion of SBML models into Python classes that can be rapidly incorporated into biomedical systems modeling projects written in Python, such as the multiscale simulation platform CompuCell3D, or used directly in Python. The Octave and MATLAB code generated by SBFC implement SBML models as a function that accepts no input and does not return any values. SBMLtoODEpy generates code that uses Python classes to create code that users write their own code to interface with. The program aims to accelerate construction of multiscale models that import and reuse published SBML models, many of which are available in the BioModels Database at https://www.ebi.ac.uk/biomodels/

In SMBLtoODEpy, each of the model components are extracted using libSBML, a software library for parsing and editing SBML models (Bornstein, Keating, Jouraku, & Hucka, 2008). The model components can be output to a JSON file. JSON is a format that is easier for users to read and directly edit than either the SBML or Python implementations of the model. The extracted model components are used to create a Python file that defines a class that implements the model. A method for the class is generated to solve the model using a wrapper for the lsoda algorithm in the SciPy Python package (Oliphant, 2007), and the NumPy Python package (Van Der Walt, Colbert, & Varoquaux, 2011) is also used. To verify that SBMLtoODEpy properly interprets SBML files and converts them into functional differential equations models, we compared the results of SBMLtoODEpy with COPASI, a graphical user interface based platform for simulating SBML models (Hoops et al., 2006), for a set of representative SBML files downloaded from the BioModels Database that were deposited for a selection of systems biology publications (Borisov et al., 2009; Guyton, Coleman, & Granger, 1972; Kerkhoven et al., 2013; Smallbone & Corfe, 2014; Waugh & Sherratt, 2006; Zi et al., 2011). This comparison served purely to verify that SBMLtoODEpy was properly setting up the equations for the model as both COPASI and SBMLtoODEpy rely on ODEPACK to solve differential equations (Hindmarsh, 1983). These files have been included in the

SBMLtoODEpy package within the sbmltoodepy/sbml_files subdirectory to serve as examples for users. In our documentation, we have provided a tutorial on how to use the SBMLtoODEpy software package.

Acknowledgments

The research results discussed in this publication were made possible in part by funding through the award for project number HR17-057, from the Oklahoma Center for the Advancement of Science and Technology.

References

Borisov, N., Aksamitiene, E., Kiyatkin, A., Legewie, S., Berkhout, J., Maiwald, T., Kaimachnikov, N. P., et al. (2009). Systems-level interactions between insulin–EGF networks amplify mitogenic signaling. *Molecular Systems Biology*, *5*, 256. doi:10.1038/msb.2009.19

Bornstein, B. J., Keating, S. M., Jouraku, A., & Hucka, M. (2008). LibSBML: An API library for SBML. *Bioinformatics*, *24*(6), 880–881. doi:10.1093/bioinformatics/btn051

Chelliah, V., Juty, N., Ajmera, I., Ali, R., Dumousseau, M., Glont, M., Hucka, M., et al. (2015). BioModels: Ten-year anniversary. *Nucleic Acids Research*, *43*(D1), D542–D548. doi:10.1093/nar/gku1181

Finney, A., & Hucka, M. (2003). Systems biology markup language: Level 2 and beyond. *Biochemical Society Transactions*. doi:10.1042/bst0311472

Glont, M., Nguyen, T. V. N., Graesslin, M., Hälke, R., Ali, R., Schramm, J., Wimalaratne, S. M., et al. (2018). BioModels: Expanding horizons to include more modelling approaches and formats. *Nucleic Acids Research*, *46*(D1), D1248–D1253. doi:10.1093/nar/gkx1023

Guyton, A. C., Coleman, T. G., & Granger, H. J. (1972). Circulation: Overall regulation. *Annual Review of Physiology*, *34*, 13–44. doi:10.1146/annurev.ph.34.030172.000305

Hindmarsh, A. C. (1983). ODEPACK, a systematized collection of ODE solvers. In R. S. Stepleman, M. Carver, R. Peskin, W. F. Ames, & R. Vichnevetsky (Eds.), *Scientific Computing: Applications of Mathematics and Computing to the Physical Sciences* (pp. 55–64). New York: North-Holland. Retrieved from https://computing.llnl.gov/casc/nsde/pubs/u88007.pdf

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., et al. (2006). COPASI: a COmplex PAthway SImulator. *Bioinformatics*, *22*(24), 3067–3074. doi:10.1093/bioinformatics/btl485

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., et al. (2003). The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, *19*(4), 524–531. doi:10.1093/bioinformatics/btg015

Kerkhoven, E. J., Achcar, F., Alibu, V. P., Burchmore, R. J., Gilbert, I. H., Trybiło, M., Driessen, N. N., et al. (2013). Handling uncertainty in dynamic models: The pentose phosphate pathway in Trypanosoma brucei. *PLoS Computational Biology*, *9*(12), e1003371. doi:10.1371/journal.pcbi.1003371

Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, *9*(3), 10–20. doi:10.1109/MCSE.2007.58

Rodriguez, N., Pettit, J.-B., Dalle Pezze, P., Li, L., Henry, A., Iersel, M. P. van, Jalowicki, G., et al. (2016). The systems biology format converter. *BMC Bioinformatics*, *17*(1), 154. doi:10.1186/s12859-016-1000-2

Smallbone, K., & Corfe, B. M. (2014). A mathematical model of the colon crypt capturing compositional dynamic interactions between cell types. *International Journal of Experimental Pathology*, *95*(1), 1–7. doi:10.1111/iep.12062

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30. doi:10.1109/MCSE.2011.37

Waugh, H. V., & Sherratt, J. A. (2006). Macrophage dynamics in diabetic wound dealing. *Bulletin of Mathematical Biology*, *68*(1), 197–207. doi:10.1007/s11538-005-9022-3

Zi, Z., Feng, Z., Chapnick, D. A., Dahl, M., Deng, D., Klipp, E., Moustakas, A., et al. (2011). Quantitative analysis of transient and sustained transforming growth factor- signaling dynamics. *Molecular Systems Biology*, *7*(1), 492. doi:10.1038/msb.2011.22