

ratesb_python: A Python Package for Analyzing Rate Laws in Biological Models

Longxuan Fan¹, Joseph L. Hellerstein², and Herbert M. Sauro³

¹ Viterbi School of Engineering, University of Southern California ² eScience Institute, University of Washington ³ Department of Bioengineering, University of Washington

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Open Journals](#)

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

ratesb_python is a Python package that analyzes mechanistic models of biological systems that consist of networks of chemical reactions like $2H_2 + O_2 \rightarrow 2H_2O$ [with rate laws such as $k[h_2]^2[o_2]$ that describe the rate at which the reaction proceeds]. The package focuses on rate laws of reactions, algebraic expressions that specify the rate at which reactants (e.g., H_2, O_2) are converted into products (e.g., O_2). ratesb_python analyzes rate laws to detect errors and warnings that affect the robustness and accuracy of models that use the SBML (Systems Biology Markup Language) community standard for model descriptions ([Hucka et al., 2003](#)).

Statement of Need

Mechanistic models in systems biology are essential tools for simulating and understanding the intricacies of complex biological systems, and a wide variety of rate laws are used. One commonly used rate law is *mass action* in which the reaction rate is proportional to the product of the concentrations of the reactants. To illustrate, consider a reaction in which m molecules of A combine with n molecules of B to produce r molecules of C , or $mA + nB \rightarrow rC$. The mass action rate law is $k * [A]^m * [B]^n$, where $[x]$ is the concentration of x and k is a constant.

The ratesb_python package evaluates rate laws against a library of predefined types to identify anomalies that may compromise the accuracy of mechanistic models. This process involves categorizing rate laws based on their mathematical characteristics and examining their performance within the context of the model. Such analysis enables ratesb_python to identify potential errors and warnings, including discrepancies in reactant usage or abnormal reaction fluxes. For example, if the rate law provided for the reaction $2H_2 + O_2 \rightarrow 2H_2O$ is $k_1[H_2]^2[O_2][H_2O]^2$ (where k_1 is a constant), ratesb_python reports an error since there is no defined classification for the rate law since the products are included ($[H_2O]$).

Software Description

ratesb_python analyzes rate laws to detect errors and warns about violations of best practices. Input to ratesb_python can be a file path to a model in the SBML or Antimony ([Smith et al., 2009](#)) formats, or a string in the Antimony format. The output is text and/or python objects. Control over inputs and outputs is managed by analyzer.py.

Central to ratesb_python is the ability to classify rate laws according to widely used types such as: mass action, Michaelis-Menten, and zeroth order kinetics. ratesb_python relies heavily on approaches employed in SBMLKinetics ([Xu, 2023](#)), which uses the sympy package

38 to do symbolic analysis of rate laws. `ratesb_python` refines and extends these approaches to
39 increase the accuracy of classification and to improve performance.

40 This functionality is complemented by `custom_classifier.py`, which offers users the flexibility
41 to define and classify rate laws via a structured JSON format. This adaptability is crucial for tai-
42 loring the tool to specific research requirements, highlighting `ratesb_python`'s commitment to
43 user-defined customization. Default classifications are detailed in `default_classifier.json`.

44 Error and warning messages generated during the analysis are systematically managed within
45 `messages.json`, ensuring users are well-informed of any issues detected during the examination
46 process. The results of these analyses are succinctly presented through the `Results` class in
47 `results.py`, providing users with a clear description of the findings. Here is a summary of the
48 error and warning codes along with their descriptions:

Code	Type	Brief Description
1-2	Errors	Issues with missing rate laws or expected reactants.
1001	Warning	Numeric-only rate law.
1002	Warning	Rate law unrecognized.
1003-1004	Warning	Flux relationship issues with reactants and products.
1005	Warning	Missing boundary species reactant.
1006	Warning	Non-constant parameters in rate law.
1010	Warning	Products in irreversible reaction rate law.
1020-1022	Warning	Naming conventions for parameters not followed.
1030-1037	Warning	Issues with ordering and formatting conventions in rate laws.
1040-1044	Warning	Annotations not following recommended SBO terms.
Error and warning messages to aid in rate law analysis.		

49 **Integration with Other Tools and API Capabilities**

50 `ratesb_python` is designed as a flexible, modular API and standalone tool, enabling integration
51 with various systems biology tools to facilitate rate law analysis in biological modeling projects.
52 Its development in python ensures compatibility with prevalent scientific computing tools,
53 allowing it to be added to existing systems or tailored for specific applications. It works
54 well with tools that are widely used in the SBML community, such as `libantimony` and
55 `libsbml`. Additionally, `ratesb_python` serves an educational purpose, offering a practical
56 tool for computational biology courses where students can learn about rate law analysis by
57 interacting with and modifying the API.

58 **Future Work**

59 Future developments for `ratesb_python` include enriching the library of checks and optimizing
60 the performance of classification algorithms. The goal is to expand the tool's capabilities,
61 making it a more comprehensive resource for developers and researchers alike, and to introduce
62 customization options for error and warning management, further enhancing its utility in
63 systems biology.

64 Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P.,
65 Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D.,
66 Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., ...
67 SBML Forum, the rest of the. (2003). The systems biology markup language (SBML): a
68 medium for representation and exchange of biochemical network models. *Bioinformatics*,
69 19(4), 524–531. <https://doi.org/10.1093/bioinformatics/btg015>

- 70 Smith, L. P., Bergmann, F. T., Chandran, D., & Sauro, H. M. (2009). Antimony: A modular
71 model definition language. *Bioinformatics*, 25(18), 2452–2454. [https://doi.org/10.1093/](https://doi.org/10.1093/bioinformatics/btp401)
72 [bioinformatics/btp401](https://doi.org/10.1093/bioinformatics/btp401)
- 73 Xu, J. (2023). SBMLKinetics: A tool for annotation-independent classification of reaction
74 kinetics for SBML models. *BMC Bioinformatics*, 24(1), 248. [https://doi.org/10.1186/](https://doi.org/10.1186/s12859-023-05380-3)
75 [s12859-023-05380-3](https://doi.org/10.1186/s12859-023-05380-3)

DRAFT