bioRxiv preprint doi: https://doi.org/10.1101/2022.01.17.476618; this version posted January 20, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Symbolic Kinetic Models in Python (SKiMpy): Intuitive modeling of large-scale biological kinetic models

Authors: Daniel R. Weilandt[1,2], Pierre Salvy[1,3], Maria Masid[1], Georgios Fengos[1,4], Robin Denhardt-Erikson[1,5], Zhaleh Hosseini[1,6], Vassily Hatzimanikatis[1*]

[1]Laboratory of Computational Systems Biotechnology, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

[2]Currently at: Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, USA

[3]Currently at: Cambrium GmBH, Germany

[4]Currently at: EMBION Technologies S.A., Switzerland

[5]Currently at: LYO-X GmbH, Switzerland

[6]Currently at: MRC Toxicology Unit, University of Cambridge, Cambridge, United-Kingdom

*To whom correspondence should be addressed.

Abstract

Motivation: Large-scale kinetic models are an invaluable tool to understand the dynamic and adaptive responses of biological systems. The development and application of these models have been limited by the availability of computational tools to build and analyze large-scale models efficiently. The toolbox presented here provides the means to implement, parametrize and analyze large-scale kinetic models intuitively and efficiently.

Results: We present a Python package (SKiMpy) bridging this gap by implementing an efficient kinetic modeling toolbox for the semiautomatic generation and analysis of large-scale kinetic models for various biological domains such as signaling, gene expression, and metabolism. Furthermore, we demonstrate how this toolbox is used to parameterize kinetic models around a steady-state reference efficiently. Finally, we show how SKiMpy can imple-ment multispecies bioreactor simulations to assess biotechnological processes.

Availability: The software is available as a Python 3 package on GitHub: https://github.com/EPFL-LCSB/SKiMpy, along with adequate documentation.

Contact: vassily.hatzimanikatis@epfl.ch

bioRxiv preprint doi: https://doi.org/10.1101/2022.01.17.476618; this version posted January 20, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

1 Introduction

Organisms are complex and adaptive systems, posing a challenge when investigating their response to environmental or genetic perturbations (Kitano, 2002). In this context, large-scale kinetic models are an essential tool to understand how the underlying biochemical reaction networks respond to such perturbations (Chowdhury *et al.*, 2015). However, currently, no modeling framework allows users to build and analyze large-scale kinetic models efficiently. Therefore, we propose a novel Python toolbox that enables the user to semiautomatically reconstruct a kinetic model from a constraint-based model (Salvy *et al.*, 2019). Furthermore, we express the models in terms of symbolic expressions, allowing the straightforward implementation of various analysis methods, e.g., numerical integration of the ordinary differential equations (ODEs).

Such numerical analysis requires a set of kinetic parameters describing the individual reaction characteristics. However, as parameters from literature or databases (Schomburg *et al.*, 2013) are collected *in vitro* and often fail to capture the *in vivo* reaction kinetics (Weilandt and Hatzimanikatis, 2019), a series of methods have been developed to infer parameters from phenotypic observations (Khodayari and Maranas, 2016; Saa and Nielsen, 2016; Gonzalez *et al.*, 2007; Wang *et al.*, 2004). To this end, we here present the first open-source implementation of the ORACLE framework to efficiently generate steady-state consistent parameter sets (Wang *et al.*, 2004; Miskovic and Hatzimanikatis, 2010; Chakrabarti *et al.*, 2013; Savoglidis *et al.*, 2016; Tokic *et al.*, 2020).

2 Methods

Implementing kinetic models

The system of ordinary differential equations describing the kinetics of a biochemical reaction network can be derived directly from the mass balances of the N reactants participating in the M reactions of the network:

$$\frac{dX_i}{dt} = \sum_j^M n_{ij}\ v_j(X,p),\ \forall\ i = 1, \dots, N \qquad (1)$$

where X_i denotes the concentration of the chemical i, n_{ij} is the stoichiometric coefficient of reactant i in reaction j and $v_j(\boldsymbol{X},\boldsymbol{p})$ is the reaction rate of reaction j as a function of the concentration state variables $\boldsymbol{X} = [X_1, X_2, \dots, X_N]^T$ and K kinetic parameters $\boldsymbol{p} = [p_1, p_2, \dots, p_K]^T$. The functions $v_j(\boldsymbol{X},\boldsymbol{p})$ are the given rate laws of each reaction j. An overview of the implemented rate laws is given in Table S1. Are the reactants

bioRxiv preprint doi: https://doi.org/10.1101/2022.01.17.476618; this version posted January 20, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

distributed across multiple compartments of each reactant's mass balance is modified according to (For details, see supplementary material):

$$\frac{dX_i}{dt} = \frac{V_{Cell}}{V_i} \sum_j^M n_{ij}\, v_j(\boldsymbol{X}, \boldsymbol{p}), \;\; \forall\, i = 1, \ldots, N \quad (2)$$

where V_{Cell} is the overall cell volume and V_i is the compartment volume for concentration X_i.

Efficient steady-state consistent parametrization

To overcome the scarcity of kinetic data, SKiMpy provides the means to infer the parameters efficiently on a large scale by sampling sets of kinetic parameters consistent with steady-state physiology (Wang *et al.*, 2004; Wang and Hatzimanikatis, 2006; Miskovic and Hatzimanikatis, 2010). These parameter sets are then evaluated with respect to local stability, global stability, and relaxation time to discard unstable models and models with non-physiological dynamics.

3 Usage

This toolbox implements various methods and resources that allow the user to build and analyze large-scale kinetic models in an efficient manner with a detailed account of the implemented methods given in the supplemental material. We further provide different tutorials demonstrating the modeling capabilities (Fig 1 A-C, for details, see supplementary material)

4 Conclusion

SKiMpy enables the user to reconstruct kinetic models for large-scale biochemical reaction systems. With an extensive palette of analytic methods, the software presents a versatile platform to model biological systems, as shown with the examples given in the supplemental material with models of i) *E. coli*'s metabolism, ii) a signaling pathway, iii) synthetic gene-expression circuits, and iv) different strains in a bioreactor. Furthermore, as the software generates symbolic expressions directly available to the user, SKiMpy facilitates the implementation of novel parameter inference and consistent sampling methods. Thus, SKiMpy represents a method development platform to analyze cell dynamics and physiology on a large scale. Finally, the presented toolbox increases the accessibility of large-scale kinetic models to various biological disciplines and studies ranging from biotechnology to the medical sciences.

bioRxiv preprint doi: https://doi.org/10.1101/2022.01.17.476618; this version posted January 20, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Acknowledgments

The authors want to thank Ljubisa Miskovic for valuable discussions.

Funding

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie grant agreements No. 722287 and 675585, the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 686070 and 814408 as well as by the Swiss National Centre of Competence in Research (NCCR) Microbiomes.

Conflict of Interest: none declared.

References

Behnel,S. *et al.* (2011) Cython: The Best of Both Worlds. *Computing in Science Engineering*, **13**, 31–39.

Chakrabarti,A. *et al.* (2013) Towards kinetic modeling of genome-scale metabolic networks without sacrificing stoichiometric, thermodynamic and physiological constraints. *Biotechnol J*, **8**, 1043–57.

Chowdhury,A. *et al.* (2015) Improving prediction fidelity of cellular metabolism with kinetic descriptions. *Current Opinion in Biotechnology*, **36**, 57–64.

Gonzalez,O.R. *et al.* (2007) Parameter estimation using Simulated Annealing for S-system models of biochemical networks. *Bioinformatics*, **23**, 480–486.

Hindmarsh,A.C. *et al.* (2005) SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *Acm Transactions on Mathematical Software*,

Khodayari,A. and Maranas,C.D. (2016) A genome-scale Escherichia coli kinetic metabolic model k-ecoli457 satisfying flux data for multiple mutant strains. *Nature Communications*, **7**, 13806.

Kitano,H. (2002) Computational systems biology. *Nature*, **420**, 206–210.

Meurer,A. *et al.* (2017) SymPy: symbolic computing in Python. *PeerJ Comput. Sci.*, **3**, e103.

Miskovic,L. and Hatzimanikatis,V. (2010) Production of biofuels and biochemicals: in need of an ORACLE. *Trends in Biotechnology*, **28**, 391–397.

Saa,P.A. and Nielsen,L.K. (2016) Construction of feasible and accurate kinetic models of metabolism: A Bayesian approach. *Scientific Reports*, **6**, 29635.

bioRxiv preprint doi: https://doi.org/10.1101/2022.01.17.476618; this version posted January 20, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Salvy,P. *et al.* (2019) pyTFA and matTFA: a Python package and a Matlab toolbox for Thermodynamics-based Flux Analysis. *Bioinformatics*, **35**, 167–169.

Savoglidis,G. *et al.* (2016) A method for analysis and design of metabolism using metabolomics data and kinetic models: Application on lipidomics using a novel kinetic model of sphingolipid metabolism. *Metab Eng*, **37**, 46–62.

Schomburg,I. *et al.* (2013) BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA. *Nucleic Acids Res.*, **41**, D764-772.

Tokic,M. *et al.* (2020) Large-scale kinetic metabolic models of Pseudomonas putida KT2440 for consistent design of metabolic engineering strategies. *Biotechnology for Biofuels*, **13**, 33.

Wang,L.Q. *et al.* (2004) Metabolic control analysis under uncertainty: Framework development and case studies. *Biophysical Journal*, **87**, 3750–3763.

Wang,L.Q. and Hatzimanikatis,V. (2006) Metabolic engineering under uncertainty. I: Framework development. *Metabolic Engineering*, **8**, 133–141.

Weilandt,D.R. and Hatzimanikatis,V. (2019) Particle-Based Simulation Reveals Macromolecular Crowding Effects on the Michaelis-Menten Mechanism. *Biophysical Journal*, **117**, 355–368.

bioRxiv preprint doi: https://doi.org/10.1101/2022.01.17.476618; this version posted January 20, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

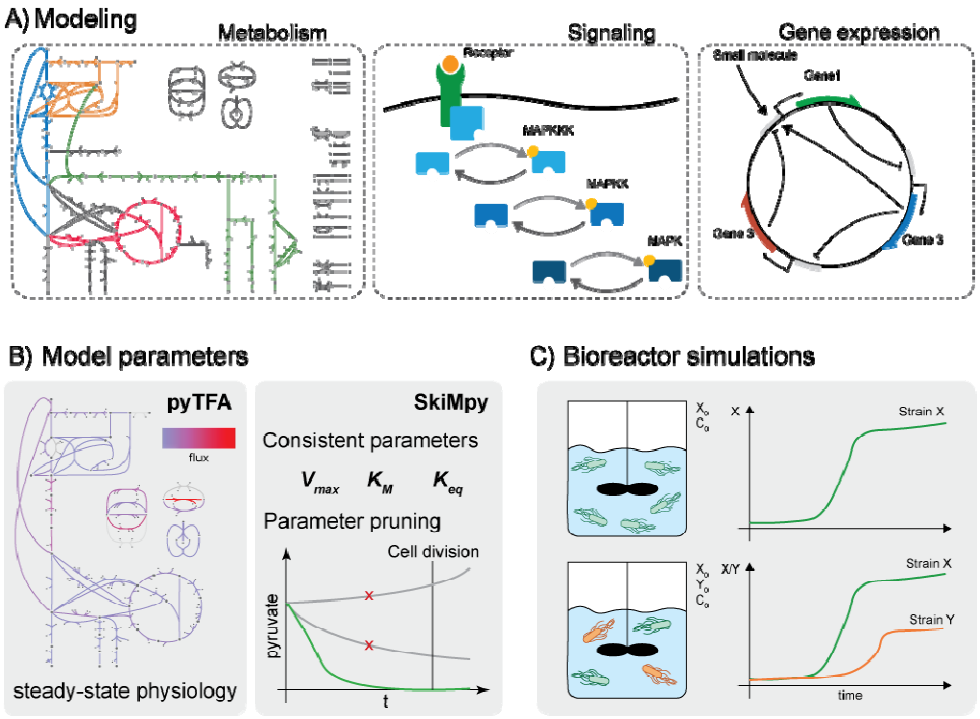

Fig. 1. Software capabilities. A) Building and simulation of different types of models B) parameterization of kinetic metabolic models around a reference steady-state C) reactor modeling and multispecies simulations