

estim8 - An FMI-compliant Python toolbox for bioprocess modeling and parameter estimation

Tobias Latour¹, Daniel Strohmeier², Michael Osthege¹, Wolfgang
Wiechert^{1,3}, and Stephan Noack¹

¹ Institute for Bio- and Geosciences (IBG-1), Forschungszentrum Jülich GmbH, Jülich, Germany
² Institute for Sustainable Hydrogen Economy (INW), Forschungszentrum Jülich GmbH, Jülich, Germany
³ Computational Systems Biotechnology, RWTH Aachen University, Aachen, Germany

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

Software

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

Editor: [Rachel Kurchin](#)

Submitted: 17 July 2025

Published: unpublished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

Modeling and simulation are indispensable tools for understanding the complex nature of biological systems and making the most of the information contained in experimental data. Functional integration of quantitative measurement data and rigorous parameter estimation are essential for creating valid, predictive models. We present *estim8*, a Python package for FMI-compliant ODE and DAE (bio)process modeling and parameter estimation. It offers convenient handling of multiple experimental replicates coupled with highly scalable solutions using federated simulation operations. The source code is available on [GitHub](#). The package is unit-tested on Windows and Linux. Thorough documentation is available under <https://estim8.readthedocs.io> including various example notebooks.

Statement of Need

Mathematical modeling has become a pivotal tool in biotechnological research and industrial bioprocess development, supporting the analysis and interpretation of complex experimental data (Fischer, 2008; Hartmann et al., 2022; Noll & Henkel, 2020; Ploch et al., 2019). While ordinary differential equations (ODEs) are commonly used to describe continuous biological systems, many biotechnological applications require differential algebraic equation (DAE) systems to handle discontinuities, discrete events, physical constraints, and embedded optimization criteria (Ploch et al., 2020).

A crucial step in the modeling workflow is parameter estimation – or in layman’s terms “fitting the model”. This step questions the theoretical understanding of the system under investigation using real data, ultimately leading to confirmation or falsification of the hypotheses put forward. Although several general-purpose software tools for model formulation, simulation and parameter estimation exist, they currently present various limitations for biotechnological applications, particularly regarding DAE support, handling of experimental replicates, and accessibility.

To address these limitations, we present *estim8*: a Python-based toolbox for simulation and parameter estimation of dynamic models. It is built on the Functional Mock-up Interface (FMI) standard (Modelica Association, 2023) and employs metaheuristic algorithms for optimization problems. *estim8* provides specialized functionality for biotechnological applications, particularly in handling experimental replicates. By supporting model definition and simulation export from various FMI-compliant third-party software, including the open source OpenModelica platform (Fritzson et al., 2020), *estim8* enables comprehensive DAE support and convenient event handling.

State of the Field

Many established tools such as pyFOOMB (Hemmerich et al., 2021), COPASI (Hoops et al., 2006), PyBioNetFit (Mitra et al., 2019), and Data2Dynamics (Raue et al., 2015) are limited to ODE systems, leaving users without native support for the DAE formulations often required in biotechnological process modeling. DAE Tools (Nikolić, 2016) and ModestPy (K. Arendt, 2018) support DAE systems but lack dedicated functionality for common biotechnological workflows such as structured handling of experimental replicates across different conditions, and therefore require substantial workarounds. The combination of AMICI (Fröhlich et al., 2021) and pyPESTO (Schälte et al., 2023) offers high-performance sensitivity analysis, but poses a relatively high entry barrier with respect to model formulation, making it less accessible to domain experts without extensive computational background. estim8 combines FMI-based DAE support, dedicated replicate handling, and an accessible Python interface to address these gaps.

Workflow

The workflow in estim8 follows a structured approach to bioprocess modeling and parameter estimation (Figure 1). Users start by developing mathematical models using third-party software that supports the FMI standard, such as OpenModelica (Fritzson et al., 2020). This open-source platform provides an interactive modeling environment with graphical features and allows models to be exported as Functional Mock-up Units (FMUs), supporting both CoSimulation and ModelExchange formats. Notably, models of the SBML (Hucka et al., 2003) standard can be translated to Modelica (Maggioli et al., 2019), further expanding the range of modeling tools supported by estim8.

The exported FMU is then loaded into estim8 using the FmuModel class. The package implements a structured data hierarchy where Experiment objects contain Measurement objects with an associated error_model and an observation_mapping. A key feature of estim8 is its comprehensive handling of biological replicates, which are crucial for the statistical quality of data from biological experiments (Casler et al., 2015). Based on a user-defined ParameterMapping, common properties of replicates and different conditions between them can be modeled by defining so-called global and local parameters. This concept utilizes redundant information in the measurement data, thereby effectively reducing the number of parameters to be estimated (Helleckes et al., 2022; Hemmerich et al., 2021; Osthege & Helleckes, 2022).

The core of estim8 is the Estimator class, which serves as a central hub for managing parameter estimation tasks. This class stores all data entered by the user and provides functions for solving optimization problems. In addition, identifiability analyses and uncertainty quantifications can be performed using profile likelihood or Monte Carlo sampling. The visualization module offers comprehensive visualization methods for analyzing simulation results, comparing model predictions with experimental data, and evaluating parameter estimation results. A thorough guideline is given by several example notebooks in our [documentation](#).

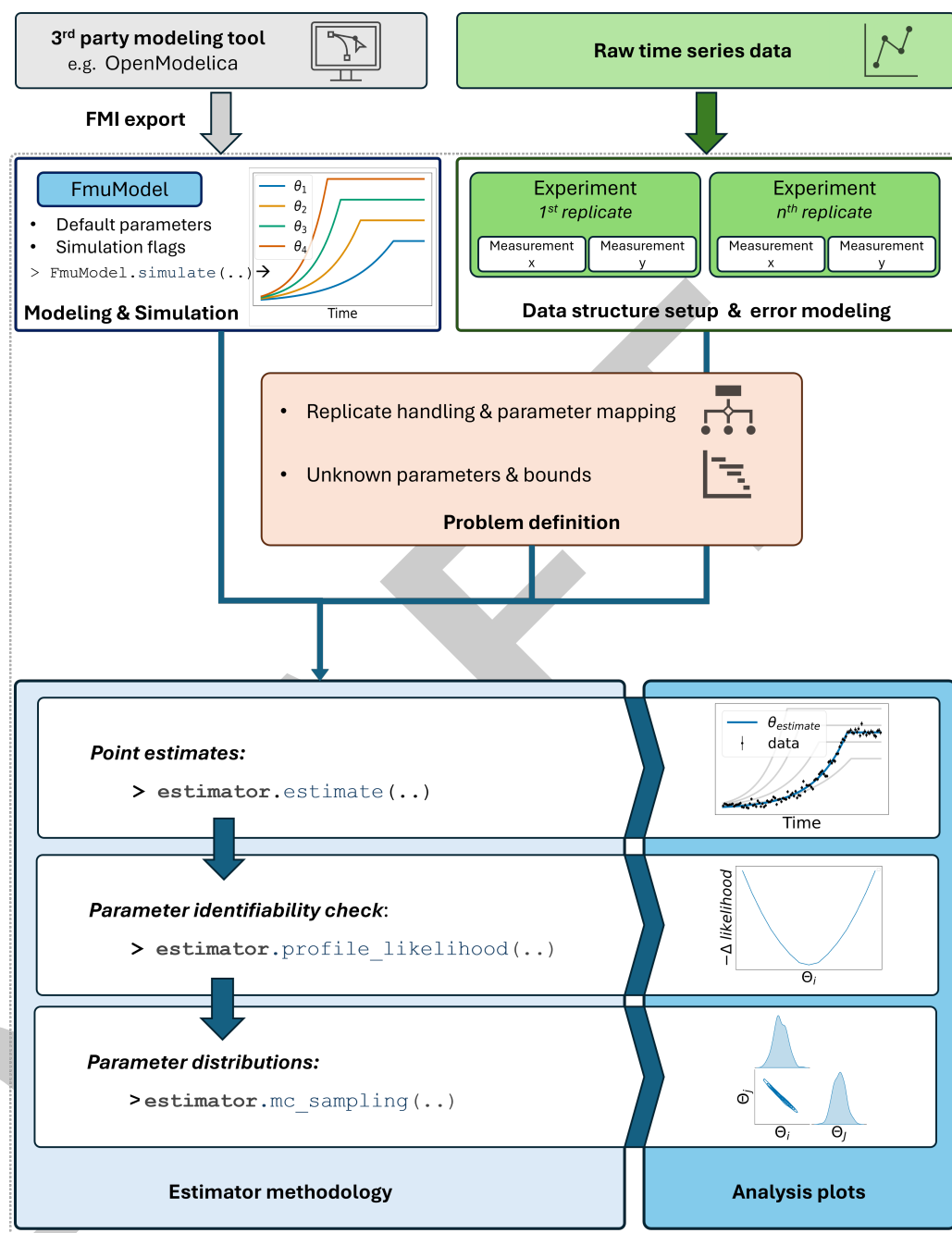


Figure 1: Schematic overview of the estim8 workflow.

Software Design

estim8 is an open-source Python package compatible and tested with Windows and Linux/Unix platforms. The modular, object-oriented architecture allows for easy expansion by new implementations, e.g. custom simulators or cost functions. At the very core, estim8 currently features interactive simulation of Functional Mock-up Units (FMUs) via the ModelExchange or CoSimulation interface in Python utilizing FMPy (Sommer, 2020). The embedded SciPy (Virtanen et al., 2020) and pygmo (Izzo & Dario, 2020) packages offer a variety of optimization algorithms that can be used for parameter estimation. Additionally, estim8 supports error propagation in measurement data and quantification of uncertainties in model predictions.

Scalability

Parameter estimation requires numerous simulation steps for evaluating a candidate solution θ by comparing the resulting model predictions to experimental data based on a statistical likelihood measure \mathcal{L} . This procedure – commonly referred to as *evaluation of the objective function* – is the time-determining step in most applications. Many population-based solvers therefore enable parallel evaluations of the objective function; the pygmo package in particular allows for highly parallelizable setups (grey parts of Figure 2).

However, the integration of experimental replicates significantly increases the number of simulations, as the objective function for a global parameter set θ_{global} is now a differentiable function resulting from the sum of replicate-specific (local) likelihood measures:

$$\mathcal{L}(\theta_{global}|y_{global}) = \sum_{i=1}^n \mathcal{L}(\theta_{local_i}|y_{local_i})$$

To this end, estim8 provides the option to use a so-called federated computing setup (Figure 2), which effectively introduces an additional parallelization layer. Using pytensor-federated (Ostehege, 2023), the computation of differentiable objective functions is distributed via gRPC streams to federated worker nodes which carry out the simulation tasks. The worker nodes can therefore be launched on different machines in a computer cluster. This allows for massive parallelization of computationally expensive model units.

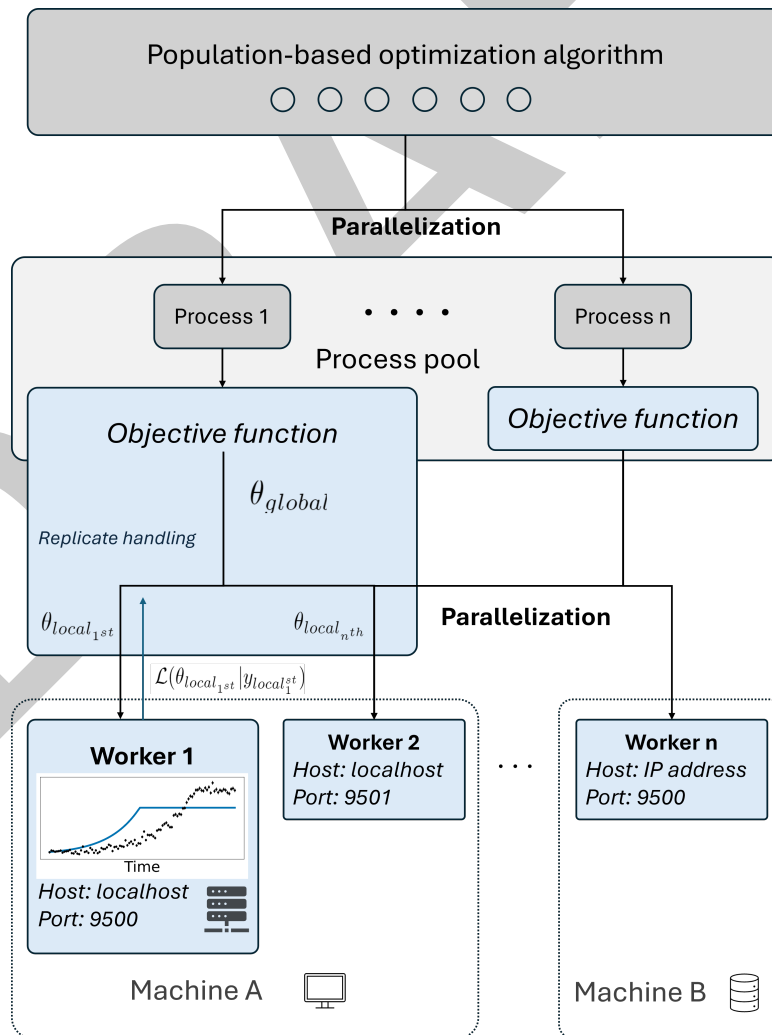


Figure 2: Federated computation setup for differentiable objective functions.

Limitations

Currently, `estim8` does not incorporate gradient-based optimization algorithms, which could enhance parameter estimation efficiency through parametric sensitivities (Villaverde et al., 2018). This capability could be implemented once OpenModelica supports FMI 3.0 (Modelica Association, 2022), providing access to adjoint derivative functions, which are essential for efficiently computing gradients in high-dimensional parameter spaces. Future developments include the integration of Bayesian optimization methods from packages like PyMC (Abril-Pla et al., 2023) and hopsy (Paul et al., 2024).

Research impact statement

`estim8` introduces a streamlined bioprocess modeling workflow that makes rigorous hypothesis testing accessible to domain experts without extensive computational background. By combining accessible Python interfaces and FMI-compliant modeling software such as OpenModelica, `estim8` allows researchers to focus on the scientific question rather than the computational implementation. With comprehensive DAE support and tailored solutions for handling experimental replicates — which are increasingly relevant in the context of laboratory automation — `estim8` enables rapid iteration through cycles of model evaluation, fitting, and falsification. This lowers the barrier for model-based reasoning in biotechnological applications, supporting faster translation of experimental observations into quantitative process understanding.

AI usage disclosure

Claude 3.5 Sonnet was occasionally used during software development for drafting and refining implementation ideas. No AI-generated code was incorporated into the software. All design decisions, code editing and reviews were conducted by the researchers. The authorship of this manuscript was conducted without the use of generative AI tools.

Author contributions

`estim8` was conceptualized by DS, SN and TL. Software developments were conducted by TL and DS, critical review of architecture decisions and contribution to the distributed computing setup by MO. The original draft was written by TL, editing and review was done by SN and MO. The work was supervised by SN and funding was acquired by SN and WW.

Acknowledgements

The authors thank Niels Hollmann and Marijke Rudolph for application tests on real-world problems as well as Mateo Herrera for checking and testing the example notebooks. Funding was received from the German Federal Ministry of Education and Research (BMBF) (grant number 031B1134A) as part of the innovation lab “AutoBioTech” within the project “Modellregion, BioRevierPLUS: BioökonomieREVIER Innovationscluster Biotechnologie & Kunststofftechnik”.

Competing interests

No competing interest is declared.

Bibliography

Abril-Pla, O., Andreani, V., Carroll, C., Dong, L., Fannesbeck, C. J., Kochurov, M., Kumar, R., Lao, J., Luhmann, C. C., Martin, O. A., Osthege, M., Vieira, R., Wiecki, T., & Zinkov, R. (2023). PyMC: a modern, and comprehensive probabilistic programming framework in Python. *PeerJ Comput. Sci.*, 9. <https://doi.org/10.7717/peerj-cs.1516>

- 150 Casler, M. D., Vermerris, W., & Dixon, R. A. (2015). Replication concepts for bioenergy
151 research experiments [Journal Article]. *BioEnergy Research*, 8(1), 1–16. <https://doi.org/10.1007/s12155-015-9580-7>
152
- 153 Fischer, H. P. (2008). Mathematical modeling of complex biological systems: From parts lists
154 to understanding systems behavior. *Alcohol Res Health*, 31(1), 49–59.
- 155 Fritzson, P., Pop, A., Abdelhak, K., Ashgar, A., Bachmann, B., Braun, W., Bouskela, D.,
156 Braun, R., Buffoni, L., Casella, F., Castro, R., Franke, R., Fritzson, D., Gebremedhin,
157 M., Heuermann, A., Lie, B., Mengist, A., Mikelsons, L., Moudgalya, K., ... Östlund,
158 P. (2020). The OpenModelica integrated environment for modeling, simulation, and
159 model-based development. *Modeling Identification and Control*, 41(4), 241–285. <https://doi.org/10.4173/mic.2020.4.1>
160
- 161 Fröhlich, F., Weindl, D., Schälte, Y., Pathirana, D., Paszkowski, Ł., Lines, G. T., Stapor, P.,
162 & Hasenauer, J. (2021). AMICI: High-performance sensitivity analysis for large ordinary
163 differential equation models. *Bioinformatics*, 37(20), 3676–3677. <https://doi.org/10.1093/bioinformatics/btab227>
164
- 165 Hartmann, F. S. F., Udugama, I. A., Seibold, G. M., Sugiyama, H., & Gernaey, K. V. (2022).
166 Digital models in biotechnology: Towards multi-scale integration and implementation.
167 *Biotechnology Advances*, 60, 108015. <https://doi.org/10.1016/j.biotechadv.2022.108015>
- 168 Helleckes, L. M., Osthege, M., Wiechert, W., von Lieres, E., & Oldiges, M. (2022). Bayesian
169 and calibration, process modeling and uncertainty quantification in biotechnology. *PLOS*
170 *Computational Biology*, 18(3), 1–46. <https://doi.org/10.1371/journal.pcbi.1009223>
- 171 Hemmerich, J., Tenhaef, N., Wiechert, W., & Noack, S. (2021). pyFOOMB: Python framework
172 for object oriented modeling of bioprocesses. *Engineering in Life Sciences*, 21(3-4), 242–257.
173 <https://doi.org/10.1002/elsc.202000088>
- 174 Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes,
175 P., & Kummer, U. (2006). COPASI—a COMplex PATHway Simulator. *Bioinformatics*,
176 22(24), 3067–3074. <https://doi.org/10.1093/bioinformatics/btl485>
- 177 Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P.,
178 Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D.,
179 Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J. H., ...
180 the, S. F. and the rest of. (2003). The systems biology markup language (SBML): A
181 medium for representation and exchange of biochemical network models [Journal Article].
182 *Bioinformatics*, 19(4), 524–531. <https://doi.org/10.1093/bioinformatics/btg015>
- 183 Izzo, F. B., & Dario. (2020). A parallel global multiobjective framework for optimization:
184 pagmo. *Journal of Open Source Software*, 5, 2338. <https://doi.org/10.21105/joss.02338>
- 185 K. Arendt, M. W., M. Jradi. (2018). ModestPy: An open-source python tool for parameter
186 estimation in functional mock-up units. *Proceedings of the American Modelica Conference*
187 *2018, Cambridge, MA, USA, October 9-10*. <https://doi.org/10.3384/ecp18154121>
- 188 Maggioli, F., Mancini, T., & Tronci, E. (2019). SBML2Modelica: Integrating biochemical
189 models within open-standard simulation ecosystems [Journal Article]. *Bioinformatics*, 36(7),
190 2165–2172. <https://doi.org/10.1093/bioinformatics/btz860>
- 191 Mitra, E. D., Suderman, R., Colvin, J., Ionkov, A., Hu, A., Sauro, H. M., Posner, R. G., &
192 Hlavacek, W. S. (2019). PyBioNetFit and the biological property specification language.
193 *iScience*, 19, 1012–1036. <https://doi.org/10.1016/j.isci.2019.08.045>
- 194 Modelica Association. (2022). *FMI 3.0 what's new* [Web Page]. <https://fmi-standard.org/docs/3.0/#fmi-whats-new>
195
- 196 Modelica Association. (2023). *Functional mock-up interface (FMI)* [Web Page]. <https://fmi-standard.org/>
197

- 198 Nikolić, DD. (2016). DAE tools: Equation-based object-oriented modelling, simulation and
199 optimisation software. *PeerJ Computer Science*, 2(e54). [https://doi.org/10.7717/peerj-](https://doi.org/10.7717/peerj-cs.54)
200 [cs.54](https://doi.org/10.7717/peerj-cs.54)
- 201 Noll, P., & Henkel, M. (2020). History and evolution of modeling in biotechnology: Modeling
202 & simulation, application and hardware performance. *Computational and Structural*
203 *Biotechnology Journal*, 18, 3309–3323. <https://doi.org/10.1016/j.csbj.2020.10.018>
- 204 Ostehege, M. (2023). *Pytensor-federated*. [https://github.com/michaelosthege/pytensor-](https://github.com/michaelosthege/pytensor-federated)
205 [federated](https://github.com/michaelosthege/pytensor-federated)
- 206 Osthege, M., & Helleckes, L. (2022). *JuBiotech/murefi: v5.1.0* (Version v5.1.0). Zenodo.
207 <https://doi.org/10.5281/zenodo.6006488>
- 208 Paul, R. D., Jadebeck, J. F., Stratmann, A., Wiechert, W., & Nöh, K. (2024). Hopsy —
209 a methods marketplace for convex polytope sampling in python. *Bioinformatics*, 40(7),
210 btae430. <https://doi.org/10.1093/bioinformatics/btae430>
- 211 Ploch, T., Lieres, E. von, Wiechert, W., Mitsos, A., & Hannemann-Tamás, R. (2020).
212 Simulation of differential-algebraic equation systems with optimization criteria embedded
213 in modelica. *Computers & Chemical Engineering*, 140, 106920. [https://doi.org/10.1016/j.](https://doi.org/10.1016/j.compchemeng.2020.106920)
214 [compchemeng.2020.106920](https://doi.org/10.1016/j.compchemeng.2020.106920)
- 215 Ploch, T., Zhao, X., Hüser, J., Lieres, E. von, Hannemann-Tamás, R., Naumann, U., Wiechert,
216 W., Mitsos, A., & Noack, S. (2019). Multiscale dynamic modeling and simulation of a
217 biorefinery. *Biotechnol. Bioeng.*, 116(10), 2561–2574. <https://doi.org/10.1002/bit.27099>
- 218 Raue, A., Steiert, B., Schelker, M., Kreutz, C., Maiwald, T., Hass, H., Vanlier, J., Tönsing,
219 C., Adlung, L., Engesser, R., Mader, W., Heinemann, T., Hasenauer, J., Schilling, M.,
220 Höfer, T., Klipp, E., Theis, F., Klingmüller, U., Schöberl, B., & Timmer, J. (2015).
221 Data2Dynamics: A modeling environment tailored to parameter estimation in dynamical
222 systems. *Bioinformatics*, 31(21), 3558–3560. [https://doi.org/10.1093/bioinformatics/](https://doi.org/10.1093/bioinformatics/btv405)
223 [btv405](https://doi.org/10.1093/bioinformatics/btv405)
- 224 Schälte, Y., Fröhlich, F., Jost, P. J., Vanhoefer, J., Pathirana, D., Stapor, P., Lakrisenko,
225 P., Wang, D., Raimúndez, E., Merkt, S., Schmiester, L., Städter, P., Grein, S., Dudkin,
226 E., Doresic, D., Weindl, D., & Hasenauer, J. (2023). pyPESTO: A modular and scalable
227 tool for parameter estimation for dynamic models. *Bioinformatics*, 39(11), btad711.
228 <https://doi.org/10.1093/bioinformatics/btad711>
- 229 Sommer, T. (2020). *FMPy*. <https://github.com/CATIA-Systems/FMPy>
- 230 Villaverde, A. F., Fröhlich, F., Weindl, D., Hasenauer, J., & Banga, J. R. (2018). Benchmarking
231 optimization methods for parameter estimation in large kinetic models. *Bioinformatics*,
232 35(5), 830–838. <https://doi.org/10.1093/bioinformatics/bty736>
- 233 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,
234 Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M.,
235 Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E.,
236 ... others. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python.
237 *Nature Methods*, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>