

jax-smfsb: A Python library for stochastic systems biology modelling and inference

Darren J. Wilkinson¹

¹ Department of Mathematical Sciences, University of Durham, UK

DOI: [10.21105/joss.07491](https://doi.org/10.21105/joss.07491)

Software

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

Editor: Julia Romanowska ↗ 

Reviewers:

- [@benlansdell](#)
- [@zeyus](#)

Submitted: 19 September 2024

Published: 05 February 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Many biological processes, and especially molecular biochemical processes, exhibit non-trivial stochasticity in their dynamical behaviour (Wilkinson, 2009). The popular textbook *Stochastic modelling for systems biology, third edition* (Wilkinson, 2018) describes the stochastic approach to modelling and simulation of biochemical processes, and how to do Bayesian inference for the parameters of such models using time course data (Golightly & Wilkinson, 2011). `jax-smfsb` provides a fast and efficient implementation of all of the algorithms described in Wilkinson (2018), able to effectively exploit multiple cores and GPUs, leading to performance suitable for the analysis of non-trivial research problems.

Statement of Need

Although there exist many tools for modelling biological network dynamics using deterministic approaches, typically based on ordinary differential equations (ODEs), there are relatively few flexible software libraries for modelling and simulation of stochastic biochemical networks, although *libRoadRunner* (Welsh et al., 2022) and *SBSCl* (Panchiwala et al., 2021) are notable examples. There are even fewer libraries for principled (fully Bayesian) inference for the parameters of such networks using data.

In addition to describing the mathematical framework for stochastic modelling, simulation, and inference, Wilkinson (2018) also describes a software implementation of all of the algorithms. The language chosen to illustrate the implementation was R (R Core Team, 2024), and the library is available as the package `smfsb` (Wilkinson, 2024). While this library is of significant pedagogical value, the overheads of dynamic interpreted languages such as R make it unsuitable for the development of high-performance codes needed for non-trivial research problems. An implementation in the compiled strongly-typed functional language Scala (Odersky et al., 2004), `scala-smfsb` (Wilkinson, 2019) partially addresses this issue, but the lack of systems biology students and researchers familiar with Scala has limited the impact of this library. More recently, a Python (Van Rossum & Drake, 2009) port of the library, `python-smfsb` (Wilkinson, 2023) has been developed, utilising the Python libraries `numpy` (Harris et al., 2020) and `scipy` (Virtanen et al., 2020). This is of significant pedagogical value, since Python has become a more popular programming language for systems biology modelling than R. Nevertheless, the performance of this library is similar to that of the R library, inadequate for serious research problems.

`jax-smfsb` addresses all of the limitations of the previously described implementations. It is essentially a port of `python-smfsb` with `numpy` and `scipy` replaced by JAX (Bradbury et al., 2018). JAX is a state-of-the-art high-performance machine learning framework that turns out to be well-suited to a range of problems in numerical, scientific and statistical computing. JAX is effectively a functional language for differentiable array processing embedded in Python, allowing just-in-time compilation and execution on modern hardware with state-of-the-art

performance. In addition to a large number of machine learning libraries based on JAX, a growing ecosystem of libraries for scientific computing is developing; see, for example, `diffraction` (Kidger, 2021), JAX-MD (Schoenholz & Cubuk, 2021), and JAX-Fluids (Bezgin et al., 2023). `jax-smfsb` adds to this ecosystem by providing tools for modelling, simulation and Bayesian inference for stochastic (biochemical) network models.

Features

Similarly to the NumPy version of the library, in addition to providing a small library of pre-defined models, and allowing direct specification of models as stochastic Petri nets using arrays, `jax-smfsb` can also parse models encoded in the Systems Biology Markup Language (SBML) (Keating et al., 2020) using `libSBML` (Bornstein et al., 2008), in addition to those encoded using the SBML-shorthand notation used in Wilkinson (2018). Exact and approximate stochastic simulation algorithms are provided for both the well-mixed and spatial (reaction-diffusion) case. Exact and approximate Bayesian inference algorithms for parameter inference are provided based on ABC (Marin et al., 2011), ABC-SMC (Toni et al., 2008) or particle MCMC (Andrieu et al., 2010) approaches.

Many demos are provided in the demos directory. These are direct equivalents to the demos provided in the `python-smfsb` library, so it is easy to compare the performance of the Python+JAX implementations to the more conventional Python+Numpy implementation. Precise timings are very hardware and problem dependent, but on a high-specification machine, speed-ups of around two orders of magnitude for computationally intensive simulation or inference tasks can be expected. On a Linux server with an Intel i7-12700 processor, the CPU-only version of `jax-smfsb` gives speedup factors relative to `python-smfsb` ranging from around 50 to around 2000 on the timing examples included in the demos directory.

References

- Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(3), 269–342. <https://doi.org/10.1111/j.1467-9868.2009.00736.x>
- Bezgin, D. A., Buhendwa, A. B., & Adams, N. A. (2023). JAX-fluids: A fully-differentiable high-order computational fluid dynamics solver for compressible two-phase flows. *Computer Physics Communications*, 282, 108527. <https://doi.org/10.1016/j.cpc.2022.108527>
- Bornstein, B. J., Keating, S. M., Jouraku, A., & Hucka, M. (2008). LibSBML: An API library for SBML. *Bioinformatics*, 24(6), 880–881. <https://doi.org/10.1093/bioinformatics/btn051>
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.3.13). <http://github.com/jax-ml/jax>
- Golightly, A., & Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus*, 1(6), 807–820. <https://doi.org/10.1098/rsfs.2011.0047>
- Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Keating, S. M., Waltemath, D., König, M., Zhang, F., Dräger, A., Chaouiya, C., Bergmann, F. T., Finney, A., Gillespie, C. S., Helikar, T., Hoops, S., Malik-Sheriff, R. S., Moodie, S.

- L., Moraru, I. I., Myers, C. J., Naldi, A., Olivier, B. G., Sahle, S., Schaff, J. C., ... Zucker, J. (2020). SBML level 3: An extensible format for the exchange and reuse of biological models. *Molecular Systems Biology*, 16(8). <https://doi.org/10.15252/msb.20199110>
- Kidger, P. (2021). *On Neural Differential Equations* [PhD thesis]. University of Oxford.
- Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2011). Approximate Bayesian computational methods. *Statistics and Computing*, 22(6), 1167–1180. <https://doi.org/10.1007/s11222-011-9288-2>
- Odersky, M., Altherr, P., Cremet, V., Emir, B., Maneth, S., Micheloud, S., Mihaylov, N., Schinz, M., Stenman, E., & Zenger, M. (2004). *An overview of the Scala programming language*.
- Panchiwala, H., Shah, S., Planatscher, H., Zakharchuk, M., König, M., & Dräger, A. (2021). The systems biology simulation core library. *Bioinformatics*, 38(3), 864–865. <https://doi.org/10.1093/bioinformatics/btab669>
- R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Schoenholz, S. S., & Cubuk, E. D. (2021). JAX-MD: A framework for differentiable physics. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12), 124016. <https://doi.org/10.1088/1742-5468/ac3ae9>
- Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M. P. H. (2008). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of The Royal Society Interface*, 6(31), 187–202. <https://doi.org/10.1098/rsif.2008.0172>
- Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace. ISBN: 1441412697
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Welsh, C., Xu, J., Smith, L., König, M., Choi, K., & Sauro, H. M. (2022). libRoadRunner 2.0: A high performance SBML simulation and analysis library. *Bioinformatics*, 39(1). <https://doi.org/10.1093/bioinformatics/btac770>
- Wilkinson, D. J. (2009). Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, 10(2), 122–133. <https://doi.org/10.1038/nrg2509>
- Wilkinson, D. J. (2018). *Stochastic modelling for systems biology, third edition*. Chapman & Hall/CRC Press. <https://doi.org/10.1201/9781351000918>
- Wilkinson, D. J. (2019). Scala-smfsb. In *GitHub repository*. GitHub. <https://github.com/darrenjw/scala-smfsb>
- Wilkinson, D. J. (2023). Python-smfsb. In *GitHub repository*. GitHub. <https://github.com/darrenjw/python-smfsb>
- Wilkinson, D. J. (2024). *Smfsb: Stochastic modelling for systems biology*. <https://doi.org/10.32614/cran.package.smfsb>