The Journal of Open Source Software

Fast and flexible simulation and parameter estimation for synthetic biology using bioscrape

Anandh Swaminathan *[1], **William Poole** †[2], **Ayush Pandey** ‡[3], **Victoria Hsiao** [4], **and Richard M Murray** [5]

1 Ghost Locomotion, Mountain View, CA, USA **2** Altos Labs, San Francisco, CA, USA **3** Control and Dynamical Systems, California Institute of Technology, Pasadena, CA, USA **4** Amyris, Emeryville, CA, USA **5** Control and Dynamical Systems and Biology and Biological Engineering, California Institute of Technology, Pasadena, CA, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

In systems and synthetic biology, it is common to build chemical reaction network (CRN) models of biochemical circuits and networks. Although automation and other high-throughput techniques have led to an abundance of data enabling data-driven quantitative modeling and parameter estimation, the intense amount of simulation needed for these methods still frequently results in a computational bottleneck. Here we present bioscrape (Bio-circuit Stochastic Single-cell Reaction Analysis and Parameter Estimation) - a Python package for fast and flexible modeling and simulation of highly customizable chemical reaction networks. Specifically, bioscrape supports deterministic and stochastic simulations, which can incorporate delay, cell growth, and cell division. All functionalities - reaction models, simulation algorithms, cell growth models, partioning models, and Bayesian inference - are implemented as interfaces in an easily extensible and modular object-oriented framework. Models can be constructed via Systems Biology Markup Language (SBML) or specified programmatically via a Python API. Simulation run times obtained with the package are comparable to those obtained using C code - this is particularly advantageous for computationally expensive applications such as Bayesian inference or simulation of cell lineages. We show the package's simulation capabilities on a variety of example simulations of stochastic gene expression. We also demonstrate the package by using it to do parameter inference on a model of integrase enzyme-mediated DNA recombination dynamics with experimental data. The bioscrape package is publicly available online (Swaminathan et al., 2022) along with more detailed documentation and examples.

Statement of need

A central theme of research in systems and synthetic biology is the quantitative predictions of the behavior of complex biological systems and efficient testing of hypotheses. Mathematical modeling and analysis tools play an integral role in such predictions and can transform the way in which we design and debug synthetic engineered biological circuits.

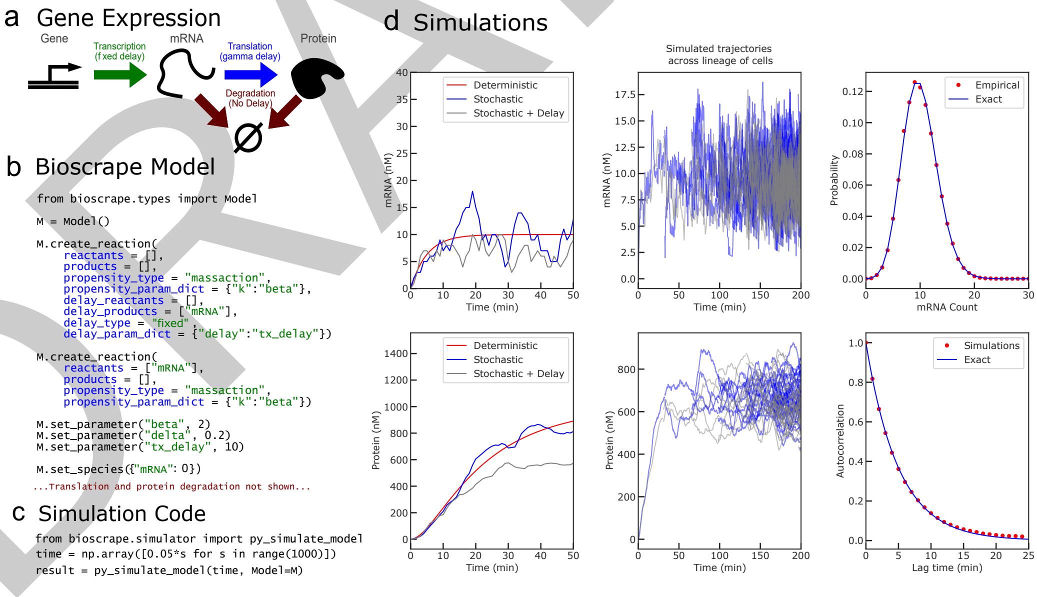
Cell growth and division are critical aspects of biological circuits which are typically represented as a dilution term in the model. However, in stochastic models, modeling the continuous dilution process with a stochastic and discrete degradation reaction might not be accurate. Moreover, the partitioning of molecules between daughter cells at cell division may introduce noise that is difficult to distinguish from other forms of noise (Huh & Paulsson, 2011). Therefore, modeling

*Co-first author
†Co-first author
‡Co-first author

cell growth as well as division and partitioning is important for investigating noise in gene expression across a lineage of cells.

Regardless of simulation framework, it is necessary to first specify the values of the parameters of each propensity function in the model along with the initial levels of the model species. In some cases, these parameters and initial conditions are experimentally known. Often, however, they have to be inferred from from biological data via a process known as parameter inference, parameter estimation, or parameter identification (Sun et al., 2012). Bayesian inference (Golightly & Wilkinson, 2011; Komorowski et al., 2009) is one of the most rigorous methods of parameter identification. It provides a posterior distribution over the parameter space so that the stochastic effects from the experimental data are modeled by the parameter distributions instead of a fixed optimal point. This gives insight into the accuracy and identifiability of the model. Also, such an approach allows for an easy comparison between different model classes using the model evidence. The drawback of these approaches is that their implementation is computationally expensive and is based on repeated forward simulations of the model within the framework of Markov chain Monte Carlo (MCMC) (Golightly & Wilkinson, 2011). Therefore, it is important to have the underlying simulations running as fast as possible in order to speed up computation time.

Once a given model is fully specified, it is then important to validate the model against additional biological data. In this workflow, it is often necessary to add or remove reactions from the model or to perform a different type of simulation. For example, one might decide that a circuit behaves too noisily for deterministic simulations and want to switch to a stochastic simulation framework. If delays are playing a significant role in the dynamics, one might want to incorporate previously unmodeled delays into the model.



b Bioscrape Model

```
from bioscrape.types import Model

M = Model()

M.create_reaction(
    reactants = [],
    products = [],
    propensity_type = "massaction",
    propensity_param_dict = {"k":"beta"},
    delay_reactants = [],
    delay_products = ["mRNA"],
    delay_type = "fixed",
    delay_param_dict = {"delay":"tx_delay"})

M.create_reaction(
    reactants = ["mRNA"],
    products = [],
    propensity_type = "massaction",
    propensity_param_dict = {"k":"beta"})

M.set_parameter("beta", 2)
M.set_parameter("delta", 0.2)
M.set_parameter("tx_delay", 10)

M.set_species({"mRNA": 0})
...Translation and protein degradation not shown...
```

c Simulation Code

```
from bioscrape.simulator import py_simulate_model
time = np.array([0.05*s for s in range(1000)])
result = py_simulate_model(time, Model=M)
```

Figure 1: (a) A simple model of gene expression with transcription, translation, mRNA degradation, and protein degradation. The quantity of the gene encoding for mRNA is considered constant and absorbed into the transcription rate β. (b) Example Python code to construct a CRN model of gene expression using Bioscrape. (c) Models constructed via SBML or the Python API can be easily simulated with results returned as a Pandas Dataframe (McKinney, 2010). (d) Deterministic and stochastic simulations (with and without delays) using Bioscrape.The empirical probability distribution and the autocorrelation function for mRNA in the stochastic simulation matches the theoretical Poisson and exponential curve respectively

The result is that a very large amount of data is needed to first parameterize and then validate models. The use of technologies for lab automation makes this data collection increasingly accessible and economical. For deterministic models, this may include data collected at many different operating conditions which can be achieved with high throughput measurement techniques involving liquid handling automation (Moore et al., 2016). For stochastic models this may include large sample sizes of single cell measurements such as flow cytometry (Sachs et al., 2005; Zechner et al., 2012) and tracking single cell lineages with fluorescent microscopy (Kretzschmar & Watt, 2012). The Python API, simulation tools, and lineage module in bioscrape provide an ideal platform for such applications.

Some popular software packages that do somewhat similar tasks as the bioscrape package are MATLAB's SimBiology toolbox (MATLAB, 2016), Stochpy (Maarleveld, 2013), COPASI (Hoops et al., 2006), and Tellurium (Medley et al., 2018). The SBML simulator libRoadRunner (Somogyi et al., 2015; Welsh et al., 2023) is the state-of-the-art in deterministic and stochastic simulations of SBML models. Bioscrape simulation performance is of the same order as libRoadRunner and only around 20-30% slower while being an order of magnitude faster than other GUI-based simulators such as MATLAB and COPASI. However, the bioscrape package provides features beyond SBML simulations as it supports fully general propensity functions, provides easy-to-use parameter identification interfaces, and allows simulation of delays, and cell populations. The target audience for bioscrape includes researchers from diverse fields such as systems biology, synthetic biology, and chemical engineering. It is also aimed as an educational tool for classes on mathematical and computational biology.

Summary of features

Figure 1 shows an example of gene expression model created and simulated stochastically and deterministically using Bioscrape.

We conclude with a list of Bioscrape features:

1. Bioscrape provides a Cython (Behnel et al., 2011) based simulator that compiles code using a C compiler to vastly increase speed. This helps assuage the computational time issues that arise in parameter estimation and stochastic simulation.
2. Kinds of possible simulations include: deterministic, stochastic, growing and dividing lineages of single cells, and stochastic simulation of delayed chemical reaction networks. A flexible easy-to-use wrapper and a Python API make it straightforward for a researcher to change their model and try simulations under diverse conditions.
3. Markov Chain Monte Carlo (MCMC) sampler based inference tools to identify parameter distributions of biological circuit models using experimental data. Bioscrape provides interfaces to easily use common biological data types such as time-series fluorescence data and flow cytometry data. The MCMC sampler is a wrapper around Python emcee (Foreman-Mackey et al., 2013).
4. Bioscrape can be used to perform local sensitivity analysis of a model to study the sensitivities of each parameter with time.

(Pandey et al., 2022) demonstrates Bioscrape's features for quantification and predictive modeling of an engineered biological system.

Acknowledgements

AS, AP, and VH were supported by the Defense Advanced Research Projects Agency (Agreement HR0011-17-2-0008). The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. AS was also supported by AFOSR grant FA9550-14-1-0060. AP was also supported by the NSF grant CBET-1903477. WP was supported by an NSF Graduate Research Fellowship (No.2017246618).

The authors acknowledge members of the Murray lab at Caltech for assistance with experiments and helpful feedback and also acknowledge all the members of the scientific community at

large who have used and provided feedback on bioscrape.

References

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The best of both worlds. *Computing in Science Engineering*, *13*(2), 31–39. https://doi.org/10.1109/mcse.2010.118

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013). emcee: The MCMC Hammer. *arXiv*, *125*, 306.

Golightly, A., & Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle markov chain monte carlo. *Interface Focus*. https://doi.org/10.1098/rsfs.2011.0047

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., & Kummer, U. (2006). COPASI—a complex pathway simulator. *Bioinformatics*, *22*(24), 3067–3074.

Huh, D., & Paulsson, J. (2011). Random partitioning of molecules at cell division. *Proceedings of the National Academy of Sciences*, *108*(36), 15004–15009. https://doi.org/10.1073/pnas.1013171108

Komorowski, M., Finkenstädt, B., Harper, C. V., & Rand, D. A. (2009). Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC Bioinformatics*, *10*(1), 343. https://doi.org/10.1186/1471-2105-10-343

Kretzschmar, K., & Watt, F. M. (2012). Lineage tracing. *Cell*, *148*(1-2), 33–45.

Maarleveld, B. G. A. B., Timo R. AND Olivier. (2013). StochPy: A comprehensive, user-friendly tool for simulating stochastic biological processes. *PLOS ONE*, *8*(11), 1–10. https://doi.org/10.1371/journal.pone.0079345

MATLAB. (2016). *Version 9.0.0 (R2016a)*. The MathWorks Inc.

McKinney, W. (2010). 10.25080/majora-92bf1922-00a. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51–56).

Medley, J. K., Choi, K., König, M., Smith, L., Gu, S., Hellerstein, J., Sealfon, S. C., & Sauro, H. M. (2018). Tellurium notebooks—an environment for reproducible dynamical modeling in systems biology. *PLoS Computational Biology*, *14*(6), e1006220.

Moore, S. J., MacDonald, J. T., Weinecke, S., Kylilis, N., Polizzi, K. M., Biedendieck, R., & Freemont, P. S. (2016). Prototyping of bacillus megaterium genetic elements through automated cell-free characterization and bayesian modelling. *bioRxiv*.

Pandey, A., Rodriguez, M. L., Poole, W., & Murray, R. M. (2022). Characterization of integrase and excisionase activity in cell-free protein expression system using a modeling and analysis pipeline. *bioRxiv*. https://doi.org/10.1101/2022.10.05.511053

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., & Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, *308*(5721), 523–529.

Somogyi, E. T., Bouteiller, J.-M., Glazier, J. A., König, M., Medley, J. K., Swat, M. H., & Sauro, H. M. (2015). libRoadRunner: A high performance SBML simulation and analysis library. *Bioinformatics*, *31*(20), 3315–3321.

Sun, J., Garibaldi, J. M., & Hodgman, C. (2012). Parameter estimation using metaheuristics in systems biology: A comprehensive review. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, *9*(1), 185–202. https://doi.org/10.1109/tcbb.2011.63

Swaminathan, A., Poole, W., Pandey, A., Hsiao, V., & Murray, R. M. (2022). Bioscrape: Biological stochastic simulation of single cell reactions and parameter estimation. In *GitHub repository*. GitHub. https://github.com/biocircuits/bioscrape/

Welsh, C., Xu, J., Smith, L., König, M., Choi, K., & Sauro, H. M. (2023). libRoadRunner 2.0: A high performance SBML simulation and analysis library. *Bioinformatics*, *39*(1), btac770.

Zechner, C., Ruess, J., Krenn, P., Pelet, S., Peter, M., Lygeros, J., & Koeppl, H. (2012). Moment-based inference predicts bimodality in transient gene expression. *Proceedings of the National Academy of Sciences*, *109*(21), 8340–8345. https://doi.org/10.1073/pnas.1200161109