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** Computation and Neural Systems, California Institute of Technology, Pasadena, 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Published: unpublished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 first show the package's simulation capabilities on a variety of example simulations of stochastic gene expression. We then further 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 (https://github.com/biocircuits/bioscrape) along with more detailed documentation and examples.

Statement of need

In the fields of systems and synthetic biology, it has become increasingly common to build mathematical models of biochemical networks. In principle, such models allow for quantitative predictions of the behavior of complex biological systems and efficient testing of hypotheses regarding how real biological networks function. Such predictions would transform the way in which we design and debug synthetic engineered biological circuits.

Biological circuits can often be noisy (Eldar & Elowitz, 2010; Elowitz et al., 2002), especially in single cells with low molecular copy numbers (Paulsson, 2005). In these cases, a stochastic model is often necessary to capture the noise characteristics of a circuit.

[*] Co-first author
[†] Co-first author
[‡] Co-first author

Stochastic simulation also allows for the inclusion of delay into chemical reactions. Processes like protein production are not instantaneous, and there is often a significant delay between when transcription of a gene is initiated and when a mature protein is produced. This type of delay can lead to non-trivial behavior such as oscillations (Stricker et al., 2008), and thus it is often important to incorporate delay into the modeling framework.

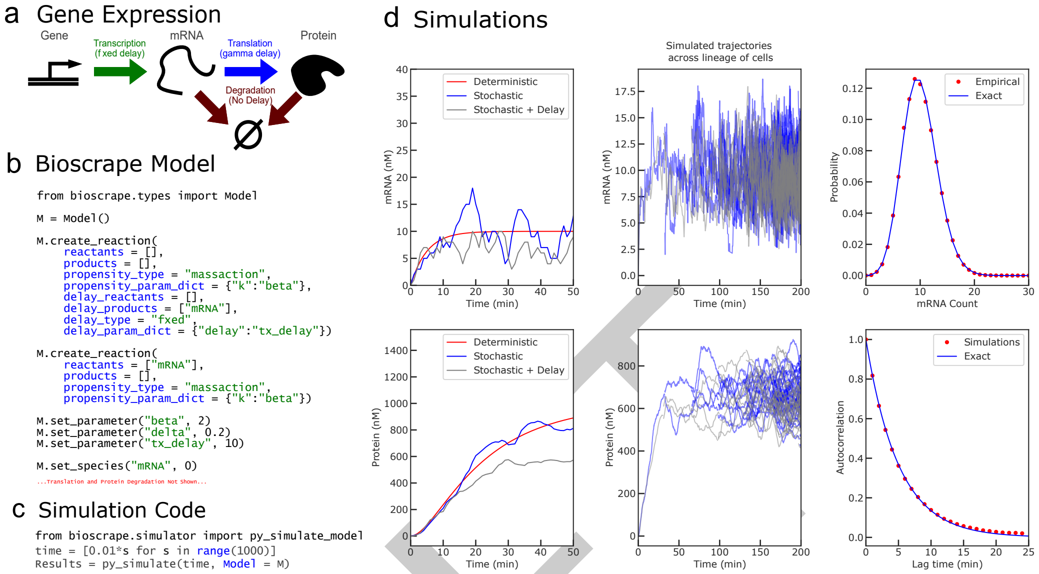
Cell growth and division are also critical aspects of biological circuits that operate in single cells. Typically, a dilution term in the model accounts for cell growth. However, in stochastic models, modeling the continuous dilution process with a stochastic and discrete degradation reaction might not be accurate. Another source of noise is the partitioning of molecules between daughter cells at cell division, which can be difficult to distinguish from other forms of noise (Huh & Paulsson, 2011). Therefore, modeling 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.

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 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).

Summary of features



```
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 = "fxed",
    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...
```

```
from bioscrape.simulator import py_simulate_model
time = [0.01*s for s in range(1000)]
Results = py_simulate(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 figure Figure 1 shows an example..

This paper presents bioscrape (Bio-circuit Stochastic Single-cell Reaction Analysis and Parameter Estimation), which is a Python package for fast and flexible modeling and simulation of biological circuits. The bioscrape package uses Cython (Behnel et al., 2011), an extension for Python 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. Bioscrape provides an object oriented framework which allows for easily customizable models that can be simulated in many different ways including deterministically, stochastically, or as growing and dividing lineages of single cells. 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. Some popular software packages that do somewhat similar tasks as the bioscrape package are MATLAB's SimBiology toolbox (MATLAB, 2016) and Stochpy (Maarleveld, 2013). However, the bioscrape package is faster, supports fully general propensity functions, and allows more kinds of simulation than these alternatives making it more flexible and more efficient than alternative packages.

Citations

Citations to entries in paper.bib should be in rMarkdown format.

If you want to cite a software repository URL (e.g. something on GitHub without a preferred citation) then you can do it with the example BibTeX entry below for (**fidgit?**).

For a quick reference, the following citation commands can be used: - `@author:2001` ->

“Author et al. (2001)” - `[@author:2001]` -> “(Author et al., 2001)” - `[@author1:2001; @author2:2001]` -> “(Author1 et al., 2001; Author2 et al., 2002)”

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.

Eldar, A., & Elowitz, M. B. (2010). Functional roles for noise in genetic circuits. *Nature*, *467*(7312), 167–173.

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, *297*(5584), 1183–1186.

Golightly, A., & Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle markov chain monte carlo. *Interface Focus*.

Huh, D., & Paulsson, J. (2011). Random partitioning of molecules at cell division. *Proceedings of the National Academy of Sciences*, *108*(36), 15004–15009.

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.

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.

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

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51–56).

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*.

Paulsson, J. (2005). Models of stochastic gene expression. *Physics of Life Reviews*, *2*(2), 157–175.

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.

Stricker, J., Cookson, S., Bennett, M. R., Mather, W. H., Tsimring, L. S., & Hasty, J. (2008). A fast, robust and tunable synthetic gene oscillator. *Nature*, *456*(7221), 516–519.

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.

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.