STReNGTHS, a Python package to model and simulate complex reaction-diffusion systems

Thibault Fillion, Francesco Piazza

The Journal of Open Source Software

STReNGTHS, a Python package to model and simulate complex reaction-diffusion systems

Thibault Fillion[1,2,5] **and Francesco Piazza**[3,4]

1 Université d'Orléans, UFR Sciences Techniques, Avenue du Parc Floral, BP 6749, 45067 Orléans, France **2** Centre de Biophysique Moléculaire (CBM), CNRS-UPR 4301, Rue Charles Sadron, 45071 Orléans, France **3** Dipartimento di fisica & Astronomia, Università di Firenze, Via G. Sansone 1, 50019 Sesto Fiorentino, Italy **4** INFN sezione di Firenze, Via G. Sansone 1, 50019 Sesto Fiorentino, Italy **5** Dipartimento di Medicina Sperimentale e Clinica, Università di Firenze, Viale Giovanni Batista Morgagni 50, 50314 Firenze, Italy

DOI: 10.21105/joss.06495

Software

- Review
- Repository
- Archive

Editor: Nikoleta Glynatsi

Reviewers:

- @parikshitbajpai
- @jakryd

Submitted: 12 December 2023
Published: 25 May 2024

Summary

STReNGTHS is an open-source Python package that provides a simple and intuitive interface for designing models of discrete 3D heterogeneous reaction-diffusion systems and simulating their trajectories. Different algorithms are available, both stochastic (exact or approximate solutions of the associated master equation) and deterministic (numerical solutions of the corresponding rate equations). The acronym stands for "Simulation and modeling Tool for REaction-diffusion Networks in Graphs and Tridimensional Heterogeneous Systems" (STReNGTHS). The simulation algorithms are interfaced through a general abstract interface, which makes it easy to extend STReNGTHS with new algorithms and other features. It is implemented in Python (standard library, Numpy (C. R. Harris et al., 2020) and Matplotlib (Hunter, 2007), as well as pytest (Krekel et al., 2004) for unit testing) and C++ (standard C++11 or later), and can be easily installed from the Python Package Index (PyPI, https://pypi.org) with (i.e.)

```
pip install strengths
```

Statement of need

Biology at the cell scale relies on complex networks of biochemical reactions, usually operating across multiple membrane-associated as well as membrane-less compartments (i.e. plasma membrane, cytoplasm, cellular organelles, stress granules, etc) and driven far from equilibrium by highly regulated species, such as nucleotides (ATP/GTP), amino acids and different ions. In order to understand the properties of such reaction-diffusion networks, and especially their role in macroscopic *emergent* phenomena, convenient, reliable and efficient modeling and simulation tools supporting heterogeneous systems are necessary. Moreover, the choice of the simulation algorithm to be used may depend on the system: deterministic approaches such as ODE integration (rate equations) are effective, but inappropriate for systems that are sensitive to fluctuations, such as systems that operate with species present at low copy numbers (e.g. certain enzymes, many mRNA species) and/or in tiny reaction volumes (i.e. within mitochondria), for which stochastic methods should be preferred. This is why one needs to have multiple, flexible approaches available. STReNGTHS provides an interface to simulate reaction-diffusion systems and manipulate their trajectories, as well as full control and access to the simulation algorithms themselves.

A reaction-diffusion system is represented by the *RDSystem* class. This defines a reaction-diffusion network, which is a set of coupled chemical transformations represented by the *RDNetwork* class, augmented with a physical space where chemical species may diffuse. The

system space is discrete and consists of a 3D mesh of individual volume elements, which we refer to as *cells*. It can be either a regular grid of cubic cells with uniform volumes, or an arbitrary network of cells with different volumes, which can be obtained by coarse-graining a cell grid.

In order to account for systems with different compartments, STReNGTHS implements the system of reaction-diffusion environments, which allows the user to define different types of cells (referred to as environments) with specific reactive and diffusive properties. Many properties, such as the initial density of species, diffusion coefficients, or reaction occurrence, can be defined environment-wise.

Importantly, species can be *chemostatted*, i.e. kept at a fixed, prescribed concentration during the simulation, globally or only in specific environments or cells. Chemostatted species allow one to model non-equilibrium conditions existing in living cells that are associated with chemical potential baths, such as in the case of the tightly regulated cytoplasmic levels of ATP or ADP.

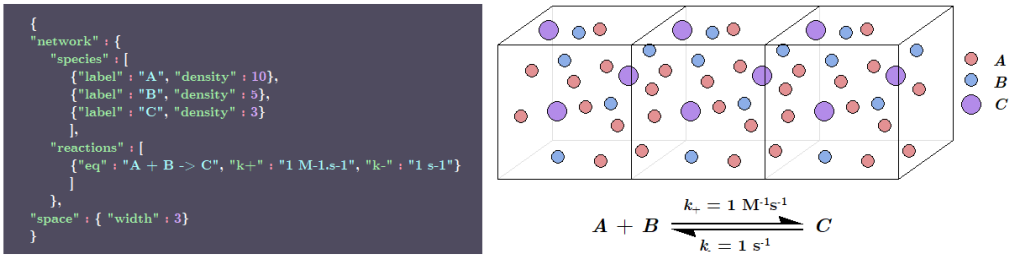

Figure 1: Definition of a simple reaction-diffusion system implementing an association reaction over 3 cells using the JSON/dictionary format. The rates used were: $k_+ = 1\ \mathrm{M}^{-1}\mathrm{s}^{-1}$, $k_- = 1\ \mathrm{s}^{-1}$.

In STReNGTHS, reaction-diffusion systems can be defined either using Python dictionaries or through JSON input files, following a specific intuitive syntax, as shown in figure 1. Simulations are handled by objects called *simulation engines*, which offer a general abstract interface for simulation algorithms. The *simulate* function wraps the engine call to run the whole simulation at once. The resulting system trajectory, which is the sequence of system states successively sampled during the simulation and the corresponding sampling times, is stored in a *RDTrajectory* object.

For now, STReNGTHS implements simulation engines for the original Gillespie algorithm (D. T. Gillespie, 1977), the τ-leap approximation to the Gillespie algorithm (D. Gillespie, 2001), and the Euler Method, operating on both grid and graph spaces, with diffusion handled according to the method described in Ref. (Bernstein, 2005) (with a slight adaptation for graph spaces).

STReNGTHS and similar tools

There already exist various computational tools to model and simulate complex reaction-diffusion systems. Some of them are general-purpose tools, while others have been designed to handle more specific systems. Existing simulation packages include:

- STEPS (Wils & De Schutter, 2009). This is a reaction-diffusion program interfaced with Python, which uses the Gillespie algorithm. It handles simulations in geometries composed of tetrahedral voxels with faces that can represent biological membranes (Wils & De Schutter, 2009).
- ReaDDy (Hoffmann et al., 2019). This is a reaction-diffusion tool with a Python interface that uses a particle-based approach. An especially interesting feature is that it can deal with complex molecule geometries and reaction patterns, such as polymer dynamics (Hoffmann et al., 2019). A Python interface is also available (Hoffmann et al., 2019).

- MesoRD (Hattne et al., 2005). This is a tool that employs a stochastic approach based on the Next Subvolume Method (Elf & Ehrenberg, 2004 December). The simulation parameters are defined through XML script files, using the System Biology Markup Language (SBML) format (Hattne et al., 2005). The software relies on Constructive Solid Geometry (CSG) to define the different reaction-diffusion compartments (Hattne et al., 2005). It comes with graphical (Windows) and command-line (Unix) user interfaces (Hattne et al., 2005).
- BioNetGen (L. A. Harris et al., 2016). This is a modeling and simulation tool that provides a rich scripting language with a rule-based approach (L. A. Harris et al., 2016). Such an approach enables one to consider systems that may be difficult to apprehend with methods requiring to define explicitly the full reaction network (such as polymerization) (L. A. Harris et al., 2016). BioNetGen supports both deterministic and stochastic methods (L. A. Harris et al., 2016).

Compared to the aforementioned tools, STReNGTHS is more rudimentary and only handles reaction networks with explicitly defined species and reactions, as opposed to pattern-based approaches or rule-based approaches used by tools such as BioNetGen (L. A. Harris et al., 2016). Still, it allows one to build in a very intuitive and user-friendly way simulations able to describe a vast range of complex systems.

So far, as opposed to Readdy, STReNGTHS only implements non-particle-based methods, similar to those proposed by the other software. However, rather than proposing only one all-purpose fitting method, STReNGTHS's approach is to provide a collection of various simulation methods, leaving the choice at the user's discretion. Moreover, simulation features can be easily extended using the simulation engine interface. In fact, STReNGTHS has been designed to be extended easily.

One of STReNGTHS' key features is the use of *reaction-diffusion environments*, which make it easy to design extremely rich system landscapes, i.e. featuring plenty of different compartments of arbitrary shape that encode physical and chemical segregation.

The use of a JSON/dictionary syntax for the definition of reaction-diffusion systems brings readability and simplicity to the workflow.

Examples

For the first example, let us consider a simple model of signal transduction, where some extracellular chemical signal is sensed by a cell, which triggers the production of a second messenger, as well as the scavenging of the signal species. The network is designed as follows: The extracellular ligand L can bind to a plasma membrane receptor R to form a complex C. This species catalyzes the conversion of the inactive second messenger X to its active form, Y. The complex C is directly converted back into R, which accounts for the internalization of the complex, the degradation of the ligand and the full recycling of the receptor. No degradation of the receptors (either through the proteasomes or lysosomes) is assumed for simplicity, although it would be straightforward to add additional reactions to implement such reaction channels (see Fig. 2 a,c). The model features 3 reaction-diffusion environments, "ext", "cyt" and "mmb", accounting, respectively, for the extracellular space, the cytoplasm and the interface between these two compartments containing the plasma membrane. We use two different system spaces. The first one is a 26×26 cell grid consisting of 1 μm^3 cubic cells, while the second one is a coarse-grained version of the first that contains only 85 nodes/cells (for 676 cells in the grid) (Fig. 2 b). The trajectory of the system state is simulated, both for the fully detailed grid and for its coarse-grained version, using the τ-leap algorithm (D. Gillespie, 2001) and the Euler method, for a total duration of 1 hour using a time step of 1 ms. The global trajectory of Y as well as its distribution at $t = 0, 100, 1500$ s are plotted in Fig. 2 (f, g).

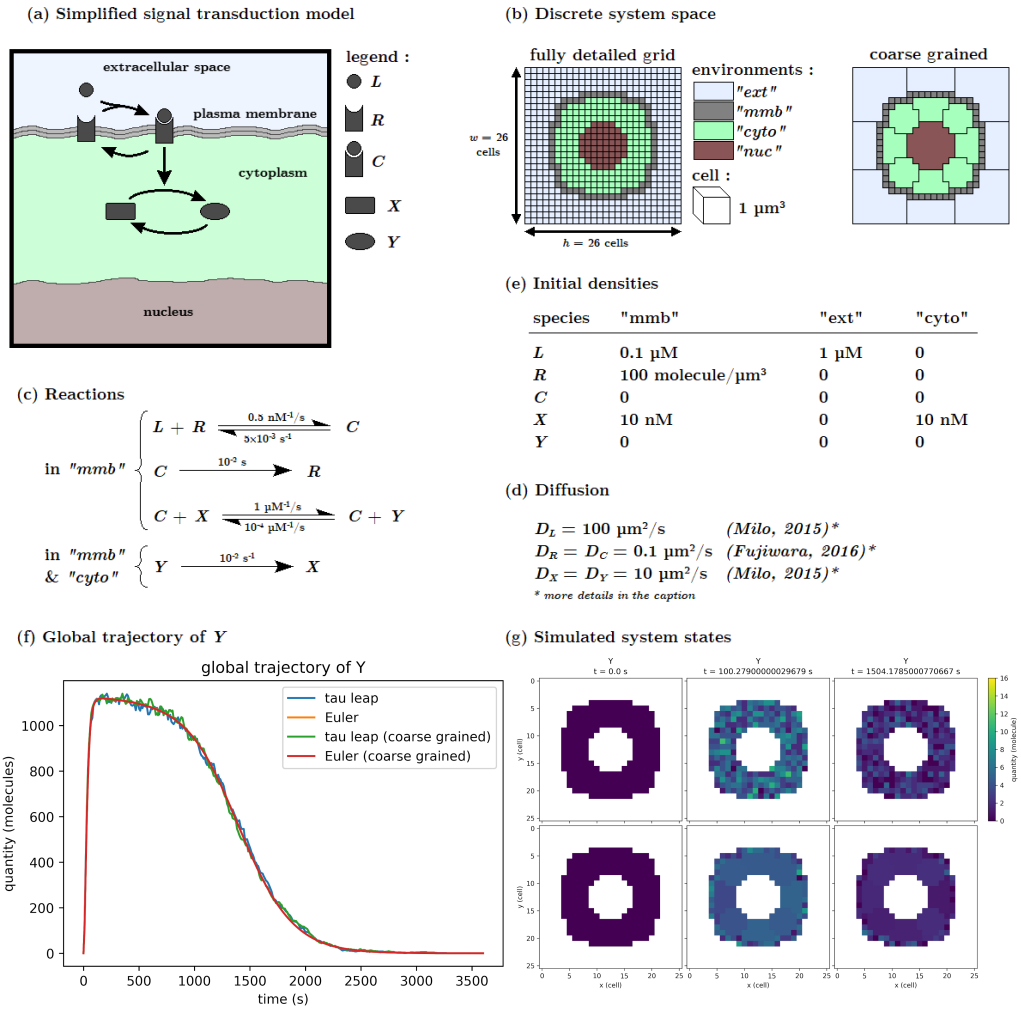

Figure 2: Example of simulation with STReNGTHS implementing a simple model of signal transduction by a single cell. (a) Schematic representation of the system. (b) Layout of the two different system spaces used, the 26×26 cell grid (left) and its coarse-grained graph version (right). (c) Set of coupled stoichiometric equations that compose the reaction network. (d) Diffusion coefficients of individual species in the different environments. (e) Initial densities of each species in the different environments. (f) Time course of the transduced signal: Global trajectory of Y obtained from the simulation using the τ-leap algorithm (D. Gillespie, 2001) and the Euler method using both system spaces (see b). (g) Distribution of the Y species at different times from the τ-leap simulations. The rates used were: $L + R \rightarrow C$: $k_1 = 0.5$ nM^{-1}s^{-1}. $C \rightarrow L + R$: $k_{-1} = 0.5 \times 10^{-3}$ s^{-1}. $C \rightarrow R$: $k_2 = 10^{-2}$ s^{-1}. $C + X \rightarrow C + Y$: $k_3 = 1$ μM^{-1}s^{-1}. $C + Y \rightarrow C + X$: $k_{-3} = 10^{-4}$ μM^{-1}s^{-1}. $Y \rightarrow X$, $k_4 = 10^{-2}$ s^{-1}. The diffusion coefficients for the different species were: $D_L = 100$ μm^{2}s^{-1} (order of magnitude for the diffusion coefficient of a protein around 30 kDa in water (Milo & Phillips, 2015)), $D_R = D_C = 0.1$ μm^{2}s^{-1} (order of magnitude for the diffusion coefficient of a transmembrane protein in a compartmented plasma membrane(Fujiwara et al., 2016)), $D_X = D_Y = 10$ μm^{2}s^{-1} (order of magnitude for the diffusion coefficient of a protein around 30 kDa in the cytosol (Milo & Phillips, 2015)).

For the second example, we consider a pattern-forming reaction-diffusion network based on the one used in the documentation of the package (Fillion & Piazza, 2024), similar to the Gray-Scott model and related reaction-diffusion schemes (McGough & Riley, 2004) (Ruijgrok & Ruijgrok, 1997), that features two competitive auto-catalytic species A and B mutually converting into each other (Fig. 3 (a)). We first simulate the evolution of the system in 1 dimension using the τ-leap algorithm (D. Gillespie, 2001). The corresponding spatiotemporal evolution of A concentration over time is reported in Fig. 3 (c) as a 2D heat map. It can be

observed how the system, starting from a homogeneous state, progressively builds up spatial reaction-diffusion patterns. We also simulate a square 100×100 cell system and plot the final distribution of A, so that the shape of the pattern can be appreciated directly (Fig. 3 (d)).

Next, we apply this model to two systems with higher complexity, where the synthesis rates of A and B vary depending on the region (Fig. 3 (b), (e), (g)). The first one (Fig. 3 (e)) represents pattern formation at the surface of a sphere, while the second one (Fig. 3 (g)) illustrates a similar phenomenon of pattern formation in a domain that takes the shape of an animal. Panels (f) and (h) in Fig. 3 demonstrate the evolution of both systems (levels of the A species) in time, highlighting the progressive evolution of the spatial patterns that form as a result of the subtle combination of the underlying auto-catalytic process with the inhomogeneous reaction landscape. Figure 3 (i) additionally illustrates how, due to the stochastic nature of the process, different patterns may arise from the same homogeneous initial state.

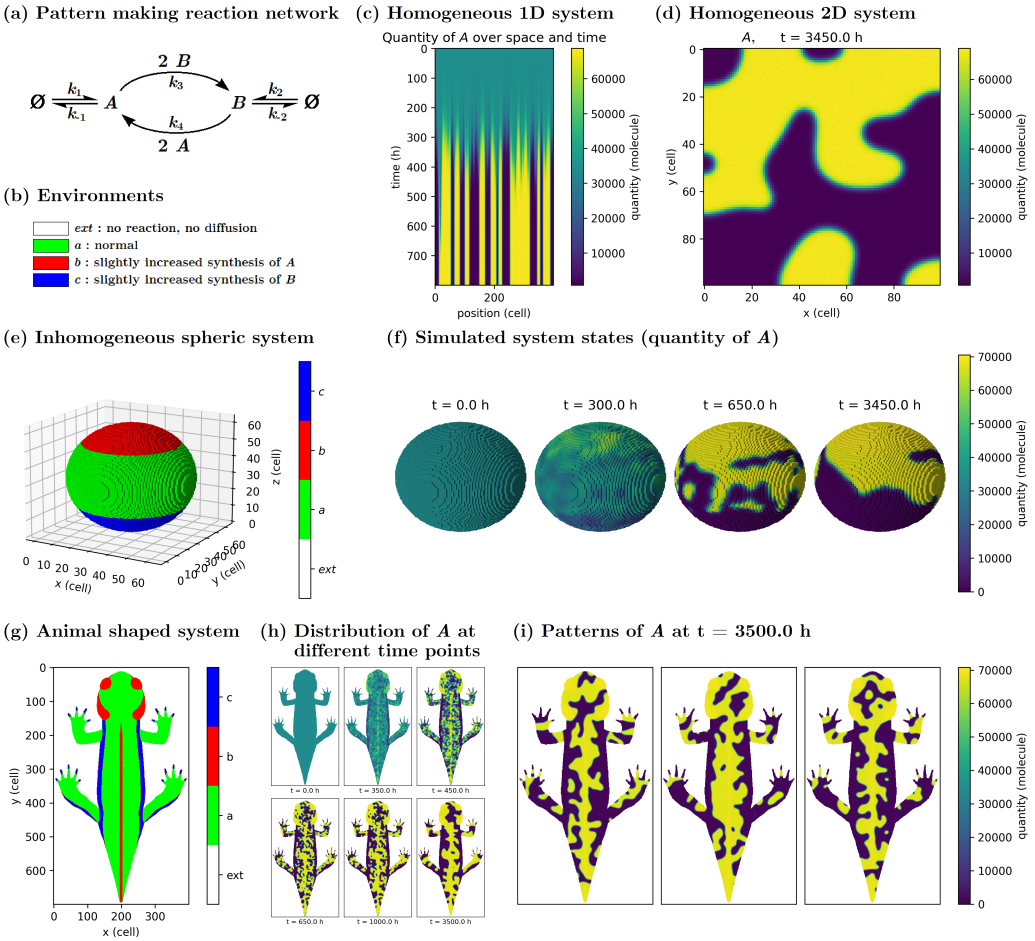

Figure 3: Example of simulations of different pattern-forming reaction-diffusion systems at increasing levels of environmental complexity. All simulations are performed with the τ-leap algorithm (D. Gillespie, 2001). (a) Description of the chemical reactions and associated rates. This reaction-diffusion network is similar to the Gray-Scott model (McGough & Riley, 2004) and the one studied by Ruijgrok and Ruijgrok (Ruijgrok & Ruijgrok, 1997). Other examples using a similar reaction-diffusion network can be found in the documentation of the package (Fillion & Piazza, 2024). (c) Evolution of a 1D system in time and space. (d) Pseudo-stationary state of a 2D system. (e) Reaction-diffusion environments for the two inhomogeneous systems. (b) Layout of a 3D system where the patterns are forming at the surface of a sphere. (f) States of the 3D system described at different time points along a stochastic trajectory (concentration of the A species). (g) Layout of a 2D system mimicking the shape of an animal. (h) States of the system (distribution of the A species) at different time points from one simulation. (i) Patterns formed at $t = 3500.0$ h resulting from 3 different stochastic simulations. The rates used were: $\emptyset \rightarrow A$: $k_1 = 10^{-4}$ molecules$\times\mu$m$^{-3}\times$h^{-1} in a and c and 1.05×10^{-4} molecules$\times\mu$m$^{-3}\times$h^{-1} in b. $A \rightarrow \emptyset$: $k_{-1} = 10^{-3}$ h^{-1}. $\emptyset \rightarrow B$: $k_2 = 10^{-4}$ molecules$\times\mu$m$^{-3}\times$h^{-1} in a and b and 1.05×10^{-4} molecules$\times\mu$m$^{-3}\times$h^{-1} in c. $B \rightarrow \emptyset$: $k_{-2} = 0.001$ h^{-1}. $A+2B \rightarrow 3B$: $k_3 = 1$ molecules$^{-2} \times \mu$m$^6\times$h^{-1}. $B + 2A \rightarrow 3A$: $k_4 = 1$ molecules$^{-2} \times \mu$m$^6\times$h^{-1}. The diffusion coefficients for the different species were: $D_A = D_B = 80$ μm^{2}h^{-1}. Reaction and diffusion rate constants are all 0 in compartment "ext".

Source code and documentation

STReNGTHS's source code and documentation are distributed under the terms of the MIT license and can be found on the dedicated GitHub repository:

https://github.com/ThibaultFillion/strengths

The documentation includes tutorials and an API reference. The tutorials demonstrate how

to define reaction-diffusion systems by taking advantage of STReNGTHS's different features (environments, chemostats, boundary conditions, etc.) as well as how to carry out simulations and post-process the trajectories. The API reference documents the exposed functions and classes.

Conclusions and perspectives

STReNGTHS is a new integrated and flexible platform for modeling and simulation of inhomogeneous reaction-diffusion systems, which aims to provide an extensible collection of stochastic and deterministic simulation engines. It has been designed to be easily integrated with new tools and we do hope it will continue to grow. Perspectives for future developments include:

- developing CPU-GPU massively parallel implementations of the existing simulation methods,
- implementing methods with dynamically adaptive time steps,
- implementing methods combining stochastic and deterministic approaches for faster simulations, and
- developing a GUI that would facilitate the design of the reaction-diffusion system layouts.

References

Bernstein, D. (2005). Simulating mesoscopic reaction-diffusion systems using the Gillespie algorithm. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, *71*, 041103. https://doi.org/10.1103/PhysRevE.71.041103

Elf, J., & Ehrenberg, M. (2004 December). Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Syst Biol (Stevenage)*, *1*(2), 230–236. https://doi.org/10.1049/sb:20045021

Fillion, T., & Piazza, F. (2024). *Building and simulating a reaction-diffusion system. Documentation for strengths*. https://strengths.readthedocs.io/en/latest/building_and_simulating_rds.html

Fujiwara, T. K., Iwasawa, K., Kalay, Z., Tsunoyama, T. A., Watanabe, Y., Umemura, Y. M., Murakoshi, H., Suzuki, K. G. N., Nemoto, Y. L., Morone, N., & Kusumi, A. (2016). Confined diffusion of transmembrane proteins and lipids induced by the same actin meshwork lining the plasma membrane. *Molecular Biology of the Cell*, *27*(7), 1101–1119. https://doi.org/10.1091/mbc.E15-04-0186

Gillespie, D. (2001). Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems. *Journal of Chemical Physics*, *115*, 1716–1733. https://doi.org/10.1063/1.1378322

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, *81*(25), 2340–2361. https://doi.org/10.1021/j100540a008

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

Harris, L. A., Hogg, J. S., Tapia, J.-J., Sekar, J. A. P., Gupta, S., Korsunsky, I., Arora, A., Barua, D., Sheehan, R. P., & Faeder, J. R. (2016). BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics*, *32*(21), 3366–3368. https://doi.org/10.1093/bioinformatics/btw469

Hattne, J., Fange, D., & Elf, J. (2005). Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics*, *21*(12), 2923–2924. https://doi.org/10.1093/bioinformatics/bti431

Hoffmann, M., Fröhner, C., & Noé, F. (2019). ReaDDy 2: Fast and flexible software framework for interacting-particle reaction dynamics. *PLoS Computational Biology*, *15*(2), e1006830. https://doi.org/10.1371/journal.pcbi.1006830

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Krekel, H., Oliveira, B., Pfannschmidt, R., Bruynooghe, F., Laugher, B., & Bruhin, F. (2004). *pytest 8.2.1*. https://github.com/pytest-dev/pytest

McGough, J. S., & Riley, K. (2004). Pattern formation in the Gray–Scott model. *Nonlinear Analysis: Real World Applications*, *5*(1), 105–121. https://doi.org/10.1016/S1468-1218(03)00020-8

Milo, R., & Phillips, R. (2015). *Cell biology by the numbers* (1st ed., p. 213). Garland Science, Taylor & Francis Group. https://doi.org/10.1201/9780429258770

Ruijgrok, T., & Ruijgrok, M. (1997). A reaction-diffusion equation for a cyclic system with three components. *Journal of Statistical Physics*, *87*, 1145–1164. https://doi.org/10.1007/BF02181277

Wils, S., & De Schutter, E. (2009). STEPS: Modeling and simulating complex reaction-diffusion systems with python. *Frontiers in Neuroinformatics*, *3*. https://doi.org/10.3389/neuro.11.015.2009