
WIREs Systems Biology and Medicine / Volume 11, Issue 6

ADVANCED REVIEW | Full Access

# Stochastic simulation algorithms for computational systems biology: Exact, approximate, and hybrid methods

Giulia Simoni, Federico Reali ... See all authors ›

First published:01 July 2019
https://doi-org.exlibris.colgate.edu/10.1002/wsbm.1459
Citations: 1

## Abstract

Nowadays, mathematical modeling is playing a key role in many different research fields. In the context of system biology, mathematical models and their associated computer simulations constitute essential tools of investigation. Among the others, they provide a way to systematically analyze systems perturbations, develop hypotheses to guide the design of new experimental tests, and ultimately assess the suitability of specific molecules as novel therapeutic targets. To these purposes, stochastic simulation algorithms (SSAs) have been introduced for numerically simulating the time evolution of a well-stirred chemically reacting system by taking proper account of the randomness inherent in such a system. In this

work, we review the main SSAs that have been introduced in the context of exact, approximate, and hybrid stochastic simulation. Specifically, we will introduce the direct method (DM), the first reaction method (FRM), the next reaction method (NRM) and the rejection-based SSA (RSSA) in the area of exact stochastic simulation. We will then present the $\tau$-leaping method and the chemical Langevin method in the area of approximate stochastic simulation and an implementation of the hybrid RSSA (HRSSA) in the context of hybrid stochastic-deterministic simulation. Finally, we will consider the model of the sphingolipid metabolism to provide an example of application of SSA to computational system biology by exemplifying how different simulation strategies may unveil different insights into the investigated biological phenomenon.

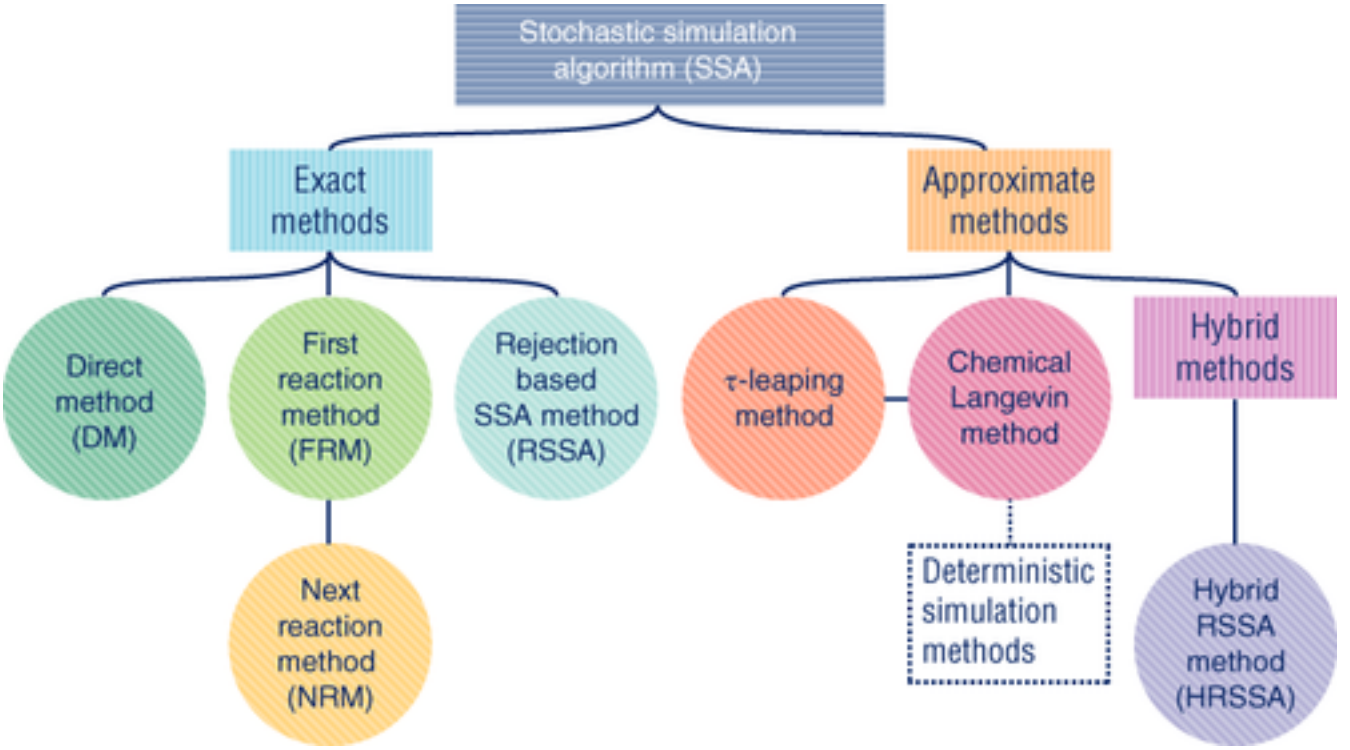
This article is categorized under:

> Models of Systems Properties and Processes > Mechanistic Models
> Analytical and Computational Methods > Computational Methods

# Abstract

A graphical representation of the simulation algorithms introduced in the review. Starting from a common root node representing a generic stochastic simulation algorithm, the methodologies differentiate in terms of accuracy and runtime according to exact and approximate methods. We depict with rectangles the algorithm classes and with circles the specific methods. Hybrid methods are here represented as a part of the approximate methods, however, they are often referred as a class of simulation algorithms itself. In the diagram, the $\tau$-leaping and the chemical Langevin methods are connected since, following the Gillespie's approach, the latter can be derived as an approximation of the former.

Analogously, the deterministic methods are connected with the chemical Langevin method. Deterministic methods are indicated with dashed lines since they are not described in detail in this review.




# 1 INTRODUCTION

The temporal evolution of a biological system is the result of complex interactions, which involve a wide variety of different biochemical components such as proteins or metabolites. One of the aims of computational system biology is to combine biological and experimental knowledge into mathematical models that are able to mimic the behavior of a biological phenomenon of interest. Once a model is defined, a computer simulation is the realization of the temporal evolution of the system by using ad hoc simulation algorithms.

Simulation algorithms are often classified as *stochastic* or *deterministic* , depending on the level of approximation they introduce into the simulation of the system dynamics. Stochastic algorithms are able to reproduce the random and discrete nature of the biological system and to reflect the two different types of noise that can affect biochemical

networks: intrinsic and extrinsic noises (Elowitz, Levine, Siggia, & Swain, 2002; Swain, Elowitz, & Siggia, 2002; Thanh, Marchetti, Reali, & Priami, 2018). Intrinsic noise is an inherent characteristic of each biological process and it is due to accidental collisions of system molecules that cause the firing of biochemical reactions. On the other hand, extrinsic noise reflects the system interactions with other quantities that are not included in the model description but are present in the environment. Certain stochastic approaches account for one or both of these types of noise, whereas deterministic approaches completely neglect their effect. However, there are many examples where stochasticity should not be ignored. This is the case of signaling and regulatory pathways, which often involve species with few replicates or a small number of particles. Indeed, when the abundance of reacting species is particularly low, the resultant fluctuations in the reaction rates may cause significant variations in the system (Gillesple, 1977; McAdams & Arkin, 1999; Pahle, 2008). Analogously, intracellular biochemical reactions often take place in very small volumes, such as in lysosomes or the Golgi apparatus, where stochastic noise may arise from countless genetic and environmental sources (Priami & Morine, 2015). In these cases, as in many others (Cai & Wang, 2007; Fisher & Henzinger, 2007; Wilkinson, 2009), stochastic simulations have been successfully applied to compute the temporal behavior of biological phenomena. In several works, the computed results have been also validated by using experimental data, as in the case of the *Escherichia coli* lactose metabolism (Halasz et al., 2007; Santillán & Mackey, 2008), the folate-mediated one-carbon metabolism (Misselbeck et al., 2017; Misselbeck, Marchetti, Priami, Stover, & Field, 2019), and cell migration (Baker, Yates, & Erban, 2010).

The class of stochastic simulation algorithms (SSAs) includes different methods that differ in accuracy and runtime (see Figure 1). Exact methods explicitly simulate each system event in an accurate and asynchronous way by proving exact realizations of the model dynamics under certain

hypotheses. These methods, however, can become prohibitively slow when applied to large biological systems, where thousands of reaction firings can happen in one time unit. To circumvent the problem, approximate methods have been introduced to decrease the runtime at the cost of sacrificing the accuracy. A popular approximation approach consists in assuming that a set of reaction events can be grouped together and fired at the same time instant, when their exact firing times are close enough according to a threshold. A drawback of approximate simulation methods is that all the model dynamics is computed according to the same approximation, while in certain cases the exactness of the simulation must be ensured at least for a subset of model reactions. For example, this happens when a part of the model represents very rare events, which would have an average firing time close to zero with an approximate approach. To take advantage of the accuracy of the exact methods and the speed of the approximate ones, hybrid methods combine the two approaches by partitioning the model reactions and by applying different simulation methods to the different partitions.

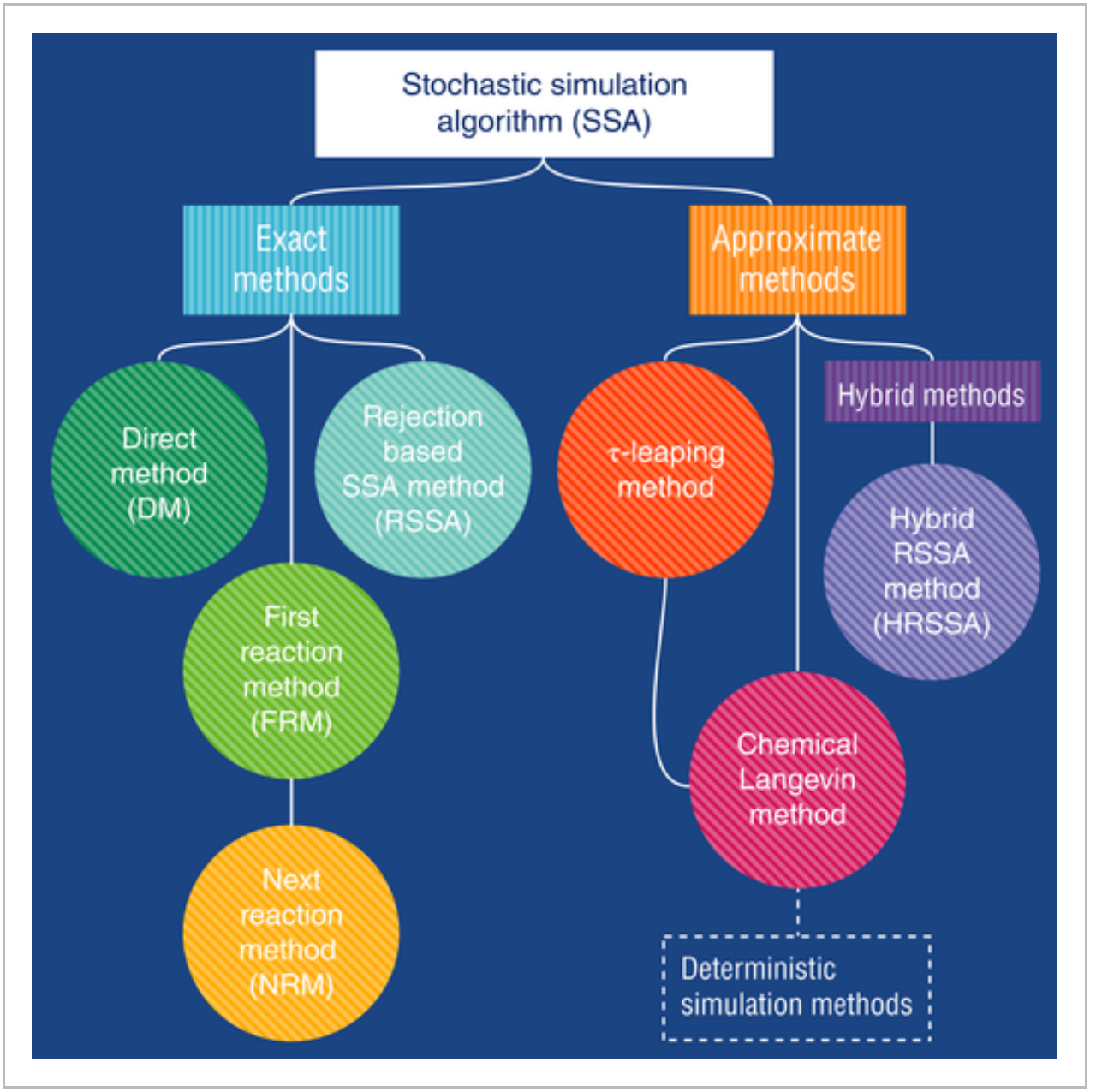



**Figure 1**

Open in figure viewer | PowerPoint

A graphical representation of the simulation algorithms introduced in the review. Starting from a common root node representing a generic stochastic simulation algorithm, the methodologies differentiate in terms of accuracy and runtime according to exact and approximate methods. We depict with rectangles the algorithm classes and with circles the specific methods. Hybrid methods are here represented as a part of the approximate methods, however, they are often referred as a class of simulation algorithms itself. In the diagram, the $\tau$-leaping and

the chemical Langevin methods are connected since, following the Gillespie's approach, the latter can be derived as an approximation of the former. Analogously, the deterministic methods are connected with the chemical Langevin method. Deterministic methods are indicated with dashed lines since they are not described in detail in this review

Deterministic simulation methods are a further approximation of stochastic algorithms in which, once an initial state is defined, the system molecules evolve continuously toward the same final state without noise. In many cases, these algorithms may provide a significant speed up compared to the exact methods, however, there are some specific system conditions, such as multistability or low species abundances, which may be not accurately reproduced by deterministic approaches. Indeed, these behaviors are the results of events that only a stochastic simulation approach is able to capture. For example, in Figure 2, we depict two simulations of the well-known Lotka-Volterra model (Lotka, 1920; Priami & Morine, 2015; Reali, Priami, & Marchetti, 2017; Volterra, 1926). This model, also known as *prey–predator model* , describes the population dynamics of two species in which one species (predator) eats the other (prey). Figure 2a is the result of a deterministic simulation. In Figure 2b, the stochastic simulation of the same model shows the extinction of the prey and the consequent disappearance of the predator. This biologically plausible scenario, which may occur when the number of preys is small, cannot be predicted by a deterministic simulation that continues oscillating indefinitely. This provides a clear evidence that the choice of the right simulation algorithm could be critical to avoid modeling artifacts. For this reason, in this review, we will focus on stochastic simulation, while deterministic simulation will be considered only when coupled with a stochastic algorithm to implement a hybrid stochastic/deterministic methodology.

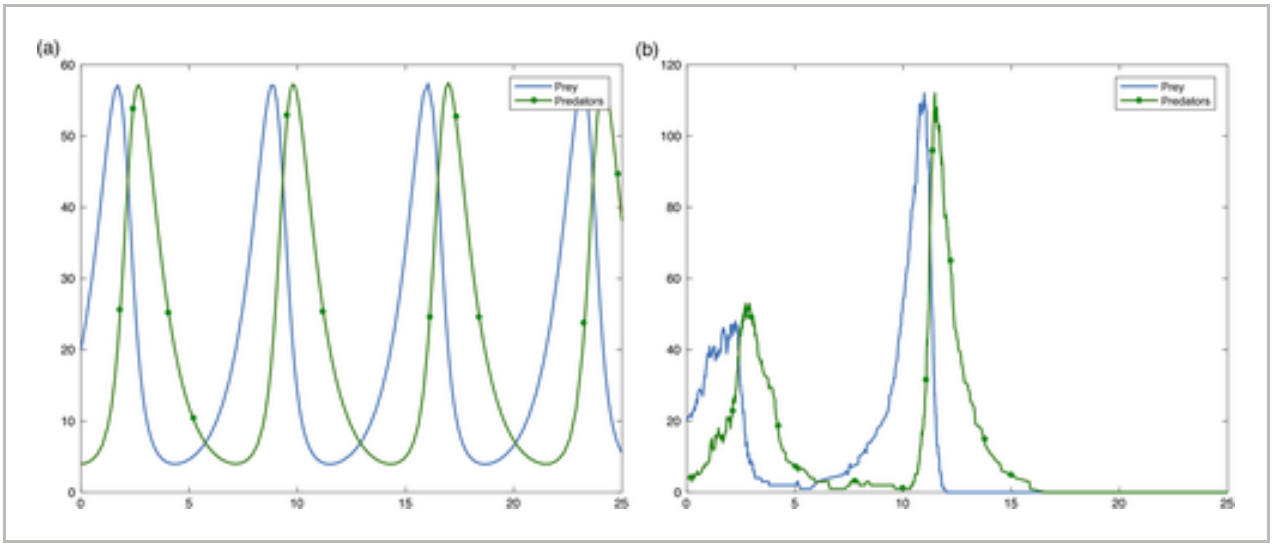



**Figure 2**

Open in figure viewer | PowerPoint

Two outcomes of the simulation of the Lotka-Volterra model with different simulation algorithms. Figure 2a Shows the results of a deterministic simulation dynamics. Figure 2b Shows the dynamics of the same model using a stochastic algorithm. *x*- axis*:* Time in absolute units. *y*- axis*:* Abundance of the species

The goal of this review is to cover some of the main algorithms that have been used in recent years for the stochastic simulation of biochemical systems. For each algorithm, a description and the complete pseudocode will be provided to present a comprehensive overview of the main stochastic algorithms and a practical guide to their implementation. We refer to Marchetti, Priami, and Thanh (2017) for the accurate benchmarking of all the simulation algorithms herein reviewed.

In addition, we exemplify how different simulation algorithms may unveil different insights into biological systems by considering ad hoc mathematical models of sphingolipid metabolism, a family of bioactive lipids that are involved in different cell regulation processes, as well as in the development of different diseases. In literature, different models have been devoted to studying sphingolipid metabolism and in particular of

ceramide. Here, we analyze how the use of three simulation algorithms have supported the different findings on two of these models (Gupta, Maurya, Merrill, Glass, & Subramaniam, 2011; Reali, Morine, et al., 2017; Thanh et al., 2018).

Section 2 introduces the reaction-based modeling framework and the basis of SSAs. In Section 2.1, exact stochastic methods are described, including the direct method (DM), a short description of the first reaction method (FRM), the next reaction method (NRM), and the rejection-based SSA (RSSA). Approximate methods are introduced in Section 2.2, including the $\tau$-leaping method and the chemical Langevin method. Section 2.3 presents the class of hybrid methods with a focus on the implementation of the hybrid RSSA (HRSSA). Finally, Section 3 presents the ceramide and sphingolipids case of study.

# 2 STOCHASTIC SIMULATION ALGORITHM

To provide a suitable and formal description of a biochemical system, we herein consider a system of biochemical reactions consisting of $N$ chemical species $S_1, \ldots, S_N$ interacting through $M$ reactions $R_1, \ldots, R_M$ in a *well-mixed environment* . Under this assumption, also referred as *spatial homogeneity hypothesis* , all the molecular species in the reaction space are homogeneously distributed and spatially indistinguishable. All the stochastic algorithms reviewed in the paper are exact only under this hypothesis.

A reaction $R_j$ has a general structure

$$v_{j1}^{-} S_1 + \ldots + v_{jN}^{-} S_N \xrightarrow{c_j} v_{j1}^{+} S_1 + \ldots + v_{jN}^{+} S_N, \tag{1}$$

where the species on the left of the arrow are *reactants* and the ones on the right are *products* . The non-negative integers $v_{ji}^{-}$ and $v_{ji}^{+}$ are the stoichiometric coefficients that indicate how many molecules of reactants and products are consumed and produced, respectively. The state change vector $\mathbf{v}_j$ accounts for the population changes in the species involved in reaction $R_j$ and its $i$ th element quantifies the change in species $S_i$ when the reaction $R_j$ is fired, that is, $\mathbf{v}_{i,j} = v_{ji}^{+} - v_{ji}^{-}$. The union of the state change vectors $\mathbf{v}_j$, $j = 1, \ldots, M$ provides the stoichiometry of the system as the so-called *stoichiometric matrix* $\mathbf{v}$ . The label $c_j$ on the arrow denotes the *stochastic reaction constant* introduced by Gillespie (1976).

The state of the system at time $t$ is represented by a vector $\mathbf{X}(t) = (X_1(t), \ldots, X_N(t))$, where $X_i(t)$ is the number of molecules of species $S_i$ in the system at time $t$ . We denote with $a_j(\mathbf{X}(t))dt$ the probability that the reaction $R_j$ fires in the next infinitesimal time $t + dt$ , given the state $\mathbf{X}(t)$ at time $t$ . The function $a_j(\mathbf{X}(t))$ is called *propensity function* , which depends on the rate constant $c_j$ and the state $\mathbf{X}(t)$.

SSA is an event-driven simulation approach that produces an exact realization of $\mathbf{X}(t)$. For each time iteration of the simulation, a reaction $R_j$ is selected to fire and to move the system to a new state according to its propensity $a_j$. Several implementations of SSA have been proposed and, according to the different levels of approximations, these implementations can be divided into: exact, approximate, and hybrid methods.

## 2.1 Exact methods

Exact stochastic methods are a class of simulation algorithms that produce an exact realization of a model dynamics. These methods are exact under the assumption that all the chemical species satisfy the spatial homogeneity hypothesis. These approaches explicitly simulate each reaction event by sampling, through random numbers, the next

reaction to fire $R_\mu$ and its firing time $\tau$ from the joint reaction probability density function (pdf)

$$p(\tau,\mu \mid \mathbf{x},t) = a_\mu(\mathbf{x})e^{-a_0(\mathbf{x})\tau}, \tag{2}$$

where $a_0$ defines the sum of all the propensities $a_\mu$, $\mu = 1, \cdots, M$ (Marchetti et al., 2017).

In the following sections, we will see four examples of exact methods that differ in the sampling step, that is, the DM, the FRM and NRM, and the RSSA. The DM splits the pdf defined in Equation 2 into two scalar pdfs. The FRM and the NRM are a variant of the DM that relies on the definition of *tentative time* . Finally, RSSA employs an acceptance–rejection sampling technique to avoid propensity updates during the simulation in specific conditions to decrease the runtime.

### 2.1.1 Direct method

The DM (Gillespie, 1976, 1977) is based on the idea of splitting the joint pdf of the two variables $\tau$ and $\mu$, defined in Equation 2, into the product of two scalar probability functions that can be independently sampled:

$$p(\tau,\mu \mid \mathbf{x},t) = p_1(\tau \mid \mathbf{x},t)p_2(\mu \mid \tau,\mathbf{x},t), \tag{3}$$

where $p_1(\tau \mid \mathbf{x}, t)$ denotes the pdf of the firing time $\tau$ and $p_2(\mu \mid \tau, \mathbf{x}, t)$ is the pdf of the reaction index $\mu$ that fires at time $t + \tau$. It can be proved (Gillespie, 1976) that the two independent pdfs can be computed as $p_2(\mu \mid \mathbf{x},t) = \frac{a_\mu}{a_0}$ and $p_1(\tau \mid \mathbf{x}, t) \sim Exp(a_0)$, the exponential probability distribution with rate $a_0$.

**Algorithm 1** Direct Method (DM)

**Input:** a biochemical reaction network of $M$ reactions in which each reaction $R_j$, $j = 1, \ldots, M$, has a state change vector $\mathbf{v}_j$ and a propensity formula $a_j$, the initial state $\mathbf{x}_0$ at time 0 and the simulation ending time $T_{max}$

**Output:** a trajectory $X(t)$, $0 \le t \le T_{max}$, of the biochemical reaction network

1: initialize time $t = 0$ and state $X = \mathbf{x}_0$.

2: **while** ($t < T_{max}$) **do.**

3: set $a_0 = 0$.

4: **for all** (reaction $R_j$) **do.**

5: compute $a_j$.

6: update $a_0 = a_0 + a_j$.

7: **end for.**

8: generate two random numbers $r_1, r_2$ in $\mathrm{U}(0,1)$.

9: select $R_\mu$ with the smallest index $\mu$ such that $\sum_{j=1}^{\mu} a_j \ge r_1 a_0$.

10: compute $\tau = \frac{1}{a_0} \ln(1/r_2)$.

11: update $X = X + \mathbf{v}_\mu$.

Algorithm 1 outlines the details of the DM. Briefly, in each iteration, the DM algorithm:

1. Compute reaction propensities in the current state and the sum of all propensities $a_0 = \sum_{\mu=1}^{M} a_\mu$ (steps 4–7).

2. Select the next reaction index $\mu$ such that it is the smallest index to satisfy $\sum_{j=1}^{\mu} \frac{a_j}{a_0} \geq r_1$, where $r_1$ is a uniformly distributed random number in the interval (0, 1) (Step 9).
3. Generate the firing time $\tau$ as $\frac{1}{a_0} ln\left(\frac{1}{r_2}\right)$, where $r_2$ is a uniformly distributed random number in the interval (0, 1) (Step 10).
4. Fire the selected reaction $R_\mu$, by updating the system state according to the stoichiometry of the reaction (Step 11).
5. Increment the simulation time by $\tau$ (Step 12).

It is important to note that the generation of random numbers significantly affects the performance of the algorithm. Indeed, the more random numbers are generated, the worse is the impact on the performance of the algorithm. In this case, the DM requires two random numbers for each iteration, one to select the reaction to fire, and one to select the firing time. We refer to (Marchetti et al., 2017; Priami & Morine, 2015) for a comprehensive discussion on the topic.

In the literature, different improvements of the DM have been developed with the goal of speeding up the selection of the next reaction firing. Among them, certain approaches sort the reactions according to the number of firing events (Cao, Li, & Petzold, 2004; McCollum, Peterson, Cox, Simpson, & Samatova, 2006), other methods use advanced search techniques, including the multilevel search (Mauch & Stalzer, 2011), the tree-based search (H. Li & Petzold, 2006; Thanh & Zunino, 2012, 2014) or the composition-rejection search (Schulze, 2008; Slepoy, Thompson, & Plimpton, 2008). Other methods employ the concept of *partial propensities* (i.e., propensity per molecules of the corresponding reactants) under specific assumptions (Ramaswamy, González-Segredo, & Sbalzarini, 2009; Ramaswamy & Sbalzarini, 2010).

### 2.1.2 First and next reaction methods

The first and next reaction methods are two variants of the DM. The FRM (Gillespie, 1976, 1977) computes for all the reactions the so called *tentative time* , that is, the firing time assuming that no other reaction fires before, and it fires the reaction with the smallest tentative time.

In this case, $p(\tau_j \mid \mathbf{x}, t)$ is the pdf of $\tau_j$ such that $p(\tau_j \mid \mathbf{x}, t)d\tau_j$ gives the probability that reaction $R_j$ fires in the next infinitesimal time interval $[t + \tau_j, t + \tau_j + d\tau_j)$ assuming that $R_j$ is the first one to fire. The formula of $p(\tau_j \mid \mathbf{x}, t)$ follows the same structure of Equation 2:

$$p(\tau_j \mid \mathbf{x}, t) = a_j e^{-a_j \tau_j}. \tag{4}$$

Thus, Equation 4 shows that the tentative time $\tau_j$ of a reaction $R_j$ is exponentially distributed with rate $a_j$. Hence, it can be generated as

$$\tau_j = \frac{1}{a_j} \ln\left(\frac{1}{r_j}\right), \tag{5}$$

in which $r_j$ is a uniformly distributed random number in the interval (0, 1). Once the tentative time has been computed for all the model reactions, the algorithm selects the next reaction to fire $R_j$ as the one with the smallest tentative time $\tau_j = \min_{h=1,\ldots,M} \{\tau_h\}$. Here, we will not provide the details and the pseudocode for the FRM as its strategy, based on the concept of tentative time, has been further improved in the NRM (Gibson & Bruck, 2000; Y. Li & Hu, 2015; Lok & Brent, 2005; Sanft & Othmer, 2015). The NRM improves the FRM by adopting a different strategy to generate random numbers and by storing and accessing the propensities and the

tentative times in suitable data structures. NRM stores the reaction dependencies, that is, those reaction whose propensities change when a reaction is fired, in a graph *G* called *reaction dependency graph* . In this way, NRM reduces significantly the runtime by identifying and updating the propensities of only those reactions that depend on the reaction firing, without updating the remaining unaffected propensities. Moreover, the simulation time is further speeded up by generating a new tentative time only for the reaction firing, while the remaining tentative times are updated and reused without generating new random numbers. In addition, these putative firing times of reactions are stored and retrieved from an efficient data structure, called *binary heap* .

**Algorithm 2** Next Reaction Method

**Input:** a biochemical reaction network of $M$ reactions in which each reaction $R_j$, $j = 1, ..., M$, has a state change vector $\mathbf{v}_j$ and a propensity formula $a_j$, the initial state $\mathbf{x}_0$ at time 0 and the simulation ending time $T_{max}$

**Output:** a trajectory $X(t)$, $0 \leq t \leq T_{max}$, of the biochemical reaction network

1: initialize time $t = 0$ and state $X = \mathbf{x}_0$.

2: build the reaction dependency graph $G$.

3: **for all** (reaction $R_j$) **do.**

4: compute $a_j$.

5: generate random numbers $r_j$ in $\mathrm{U}(0, 1)$.

6: set $t_j = (1/a_j)\ln(1/r_j)$.

7: **end for.**

8: build the binary heap $H$ for the tentative times $t_j$, $j = 1, ..., M$.

9: **while** ($t < T_{max}$) **do.**

10: extract the node with the smallest time $t_\mu$ and reaction $R_\mu$ from heap H.

11: set $t = t + t_\mu$.

The complete NRM algorithm is outlined in Algorithm 2. We refer to (Gibson & Bruck, 2000; Y. Li & Hu, 2015; Lok & Brent, 2005; Marchetti et al., 2017; Sanft & Othmer, 2015) for a more formal mathematical description of all the algorithm steps. Briefly, the NRM proceeds through the following steps:

1. Initialize algorithm data structures (steps 1–8);.
2. Extract from the heap $H$ the reaction $R_\mu$ that has the smallest time $t_\mu$ (Step 10).
3. Advance the simulation time as $t = t + t_\mu$ and update the state as $X = X + \mathbf{v}_\mu$ (steps 11–12).
4. Compute the new propensities $a_j^{new}$ for all the reactions $R_j$ that are affected by the state modification, that is, depend on reaction $R_\mu$ (Step 14).
5. Generate a new tentative time for $R_\mu$ (Step 19).
6. Update the tentative times $t_j^{new}$ for all the reactions $R_j$ without generating new random numbers (Step 16).
7. Update the heap $H$ (Step 22).

Note that the NRM is particularly parsimonious in terms of generated random numbers. In fact, after the first step, the NRM requires one random number for each iteration, considerably less than the FRM and the DM, which require $M$ and two random numbers for each iteration, respectively.

We refer to Anderson (2007) and Ethier and Kurtz (1986) for an equivalent formulation of the NRM, known as the modified NRM. This formulation combines the simulation speed of the NRM with the advantage of being easily extended to support systems with time-dependent rates.

### 2.1.3 Rejection-based SSA

All the previous exact methods compute and update the propensities at each simulation step-in a way that can become prohibitively time-consuming. To speed up the simulation time, the RSSA (Thanh, Priami, & Zunino, 2014) aims at reducing the number of propensity updates for most of the simulation steps by applying an acceptance–rejection technique based on the concept of propensity bounds. Only when the test

using the propensity bounds fails, the exact propensity must be computed. More in details, for each species $S_i$, $i = 1, \ldots, N$, RSSA bounds the population $X_i(t)$ with a *fluctuation interval* $[\underline{X_i}, \overline{X_i}]$, defined as $[\underline{X_i}, \overline{X_i}] = [(1-\delta_i)X_i(t), (1+\delta_i)X_i(t)]$ where $\delta_i$, $i = 1, \ldots, N$, is an arbitrary parameter called *fluctuation rate*. Then, for each reaction $R_j$, $j = 1, \ldots, M$, an abstract propensity bound $[\underline{a_j}, \overline{a_j}]$ is computed and used to derive the upper bound for the sum of propensities $\overline{a_0} = \sum_{j=1}^{M} \overline{a_j}(\mathbf{x}, t)$. The propensity bounds contain all the possible reaction propensities and are derived by minimizing and maximizing each propensity function $a_j$ over the fluctuation interval. For monotonic propensity functions as mass action or Michaelis–Menten kinetics, the propensity lower and upper bounds are computed as $\underline{a_j} = a_j(\underline{X})$ and $\overline{a_j} = a_j(\overline{X})$, whereas for nonmonotonic functions, they are computed by applying a numerical optimization technique.

**Algorithm 3** Rejection-Based SSA (RSSA)

**Input:** a biochemical reaction network of $M$ reactions in which each reaction $R_j$, $j = 1, ..., M$, has a state change vector $\mathbf{v}_j$ and a propensity formula $a_j$, the fluctuation rate $\delta_i$ for each species $S_i$, $i = 1, ..., N$, the initial state $\mathbf{x}_0$ at time 0 and the simulation ending time $T_{max}$

**Output:** a trajectory $X(t)$, $0 \le t \le T_{max}$, of the biochemical reaction network

1: initialize time $t = 0$ and state $X = \mathbf{x}_0$.

2: define the fluctuation interval $\left[\underline{X}, \overline{X}\right]$ of state $X$.

3: compute the propensity bounds $\overline{a_j}$ and $\underline{a_j}$ for each reaction $j = 1, ..., M$.

4: compute the total propensity upper bound $\overline{a_0} = \sum_{j=1}^{M} \overline{a_j}$.

5: **while** ($t < T_{max}$) **do.**

6: set $u = 1$.

7: set *accepted* = **false.**

8: **repeat.**

9: generate three random numbers $r_1, r_2, r_3$ in $\mathrm{U}(0,1)$.

10: select $R_\mu$ with the minimum index $\mu$ such that $\sum_{j=1}^{\mu} \overline{a_j} \ge r_1 \overline{a_0}$.

11: **if** ( [illegible] ) **then**

The complete RSSA procedure is outlined in Algorithm 3. In the following, we briefly describe how RSSA selects the next reaction to fire:

1. Select a candidate reaction $R_\mu$, where $\mu$ is the smallest reaction index such that $\sum_{j=1}^{M} \overline{a_j} \ge r_1 \overline{a_0}$ and $r_1$ is a uniformly distributed

random number in the interval (0, 1) (Step 10).

2. Validate the candidate reaction $R_\mu$ through the following rejection test:

   a. Generate $r_2$ as a uniform distributed random number in the interval (0, 1).
   b. Check whether $r_2 \leq a_\mu / \overline{a_\mu}$. If the test succeeds, accept $R_\mu$ without computing $a_\mu$.
   c. If the test fails, compute $a_\mu$ and use again $r_2$ to check if $r_2 \leq a_\mu / \overline{a_\mu}$ (steps 11–18).
   d. If $R_\mu$ is accepted, compute its firing time (Step 21). Otherwise, select and test a new reaction (steps 8–20).

To preserve the exactness of the simulation, RSSA advance the simulation time at every attempt of the rejection procedure. In details, assuming that $k$ rejections of a candidate reaction precede its acceptance, the simulation time advances by the sum of $k+1$ exponential random numbers

$$\tau = -\frac{1}{\overline{a_0}} \ln(u) = \frac{1}{\overline{a_0}} \ln \frac{1}{u_1} + \frac{1}{\overline{a_0}} \ln \frac{1}{u_2} + \cdots + \frac{1}{\overline{a_0}} \ln \frac{1}{u_{k+1}},$$

where $u = u_1 u_2 \cdots u_{k+1}$ is updated at each rejection (Step 19), and each $u_i$ is an independently and identically distributed random numbers in $\mathrm{U}(0,1)$. In the case that a candidate reaction is accepted at the first tentative, the simulation time advances as the other exact methods. We refer to Marchetti et al. (2017)and Thanh et al. (2014) for the proof of exactness of the selection procedure. After firing the accepted reaction, RSSA controls if the new system state is inside the fluctuation interval. When this happens, the algorithm does not need to recompute the propensity bounds to perform the next selection step. On the contrary,

when the new system state falls outside the fluctuation interval, a new interval is defined and all the propensities bounds must be updated (steps 24–28).
We note that RSSA requires to generate three random numbers at each rejection test, which is more than the DM and the NRM. However, this additional computational cost is largely compensated by the reduction in the number of propensity updates. We refer to Marchetti et al. (2017) and Thanh et al. (2014)) for a *in-depth* analysis of the performance of RSSA.

The successive versions of RSSA (Thanh & Priami, 2015; Thanh, Zunino, & Priami, 2015, 2017a, 2017b) improve the simulation performance by using search techniques, such as tree-based search or composition-rejection search, as the one we discussed in Section 2.1. In Thanh et al. (2018), RSSA is extended to account for the biological extrinsic noise.

## 2.2 Approximate methods

Exact methods provide the most accurate realization of the model dynamics by simulating each system event one after the other. However, this approach can become prohibitively slow when applied to large biological systems, where thousands of reactions can fire for each time step. This computational challenge sets the basis for the development of approximate algorithms that have been introduced to improve the simulation efficiency by decreasing the runtime, at the cost of reducing the accuracy. In addition to considering the spatial homogeneity hypothesis, the approximate methods introduce a new level of approximation by assuming that a set of reaction events can be fired at the same time instant, when their exact firing times are close enough according to a threshold. This idea is the basis of the $\tau$-leaping method that is further approximated with the chemical Langevin method. Note that, for approximate methods, the lower number of iterations significantly reduces the impact of generating random numbers on the

overall performance of the simulation strategy. Therefore, in the following sections, we omit a detailed discussion on the number of random numbers required at each iteration.

Another common approach to speed up the simulation is to reduce/simplify the model, rather than using an approximate simulation algorithm. This approach is popular in multiscale systems, where species abundances or kinetic rates may vary of orders of magnitude, due to the different scales of the models. The multiscale nature of these systems introduces many sources of variability and several computational challenges that the model reduction may ease. Kang and Kurtz presented in (Kang & Kurtz, 2013) how, under specific conditions, a multiscale system can be approximated by individually considering the two sources of scale variation, namely, the species abundances and the kinetic rates. However, the model reduction for multiscale systems goes beyond the scope of this review. We refer to Hepp, Gupta, and Khammash (2015) and Kang and Kurtz (2013) for comprehensive analyses of the topic.

### 2.2.1 $\tau$-leaping method

The $\tau$-leaping method (Gillespie, 2001) approximates each simulation step by firing simultaneously a group of reactions in a time interval, avoiding the generation of a separate reaction event for each individual reaction. The simulation time in the $\tau$-leaping method is discretized into time intervals of length $\tau$, which are called the *leap* time and are adaptively defined during the simulation. The value for $\tau$ must fulfill the so-called *leap condition* , that is, it must be small enough to ensure that no significant change in reaction propensities occurs during the time interval $[t , t + \tau)$. In this way, the propensity of each reaction $R_j$ can be approximate as a constant value $a_j(\mathbf{x})$ during the time interval, given $\mathbf{x} = X(t)$. Thus, the probability that $k_j$ reactions fire during $\tau$ follows a Poisson distribution given by

$$\mathbb{P}\{k_j \mid \tau, \mathbf{x}, t\} = \frac{(a_j(\mathbf{x})\tau)^{k_j}}{k_j!} e^{-a_j(\mathbf{x})\tau}. \tag{6}$$

Basically, at each simulation step, a Poisson-distributed random number $k_j \sim Poi(a_j(\mathbf{x})\tau)$ is drawn for each reaction $R_\mu$ and the system state is updated according to

$$X(t+\tau) = \mathbf{x} + \sum_{j=1}^{M} k_j \mathbf{v}_j = \mathbf{x} + \sum_{j=1}^{M} Poi\left(a_j(\mathbf{x})\tau\right)\mathbf{v}_j. \tag{7}$$

The $\tau$-leaping method may provide a significant speed gain, since many single reaction events can be leaped over, but there are several problems for its practical implementation. First, the efficiency and accuracy of the $\tau$-leaping method strongly depends on the selection of $\tau$ that satisfies the leap condition (Anderson, 2008; Cao, Gillespie, & Petzold, 2006; Gillespie & Petzold, 2003; Moraes, Tempone, & Vilanova, 2014). Second, given a $\tau$ that satisfies the leap condition, the procedure must ensure that only few reactions fire together, thereby, avoiding reaching a negative population of reactant species (Cao, Gillespie, & Petzold, 2005). Third, the procedure needs a robust condition to smoothly switch the $\tau$-leaping to the exact simulation when $\tau$ is very small (Anderson, Higham, & Sun, 2016; Cao & Petzold, 2005; Harris & Clancy, 2006; Moraes, Tempone, & Vilanova, 2015).

Algorithm 4 outlines the details of the $\tau$-leaping simulation. For each iteration, the following steps must be performed:

1. Determine $\tau$ satisfying the leap condition (Step 5).

2. Compare the leap time $\tau$ with the threshold $k/a_0$ (Step 8).

    a. If $\tau > k/a_0$, generate $M$ random number $k_j$ from the Poisson distribution $\mathtt{Poi}\left(a_j(\mathbf{x})\tau\right)$ (Step 9) and update the system state (Step 10) according to Equation 7 and the simulation time (Step 11).
    b. If $\tau \leq k/a_0$, apply an exact method to perform a predefined number of simulation steps (Step 13).

3. Check if there are species with negative populations after the application of the leap procedure. If this happens, reject the new system state and the new time step to compute a new leap procedure with a reduced $\tau$ (steps 15–19).

**Algorithm 4** $\tau$-Leaping Method

**Input:** a biochemical reaction network of $M$ reactions in which each reaction $R_j$, $j = 1, ..., M$, has a state change vector $\mathbf{v}_j$ and a propensity formula $a_j$, the initial state $\mathbf{x}_0$ at time 0, the simulation ending time $T_{max}$, the error control parameter $0 < \varepsilon \ll 1$, the reduction factor $\alpha < 1$, the threshold parameter $k$ and the number $p$ of exact SSA steps to compute.

**Output:** a trajectory $X(t)$, $0 \le t \le T_{max}$, of the biochemical reaction network.

1: initialize time $t = 0$ and state $X = \mathbf{x}_0$.

2: **while** ($t < T_{max}$) **do.**

3: compute $a_j$ for each reaction $R_j$ with $j = 1, ..., M$ and $a_0 = \sum_{j=1}^{M} a_j$.

4: set *threshold* = $k / a_0$.

5: determine $\tau$ satisfying the leap condition with a leap selection procedure.

6: **repeat.**

7: set *acceptedLeap* = **true.**

8: **if** ($\tau >$ *threshold*) **then.**

9: generate $M$ random numbers $k_j$ in $\mathrm{Poi}\left(a_j(\mathbf{x})\tau\right)$ with $j = 1, ..., M$.

10: update $X = X + \sum_{j=1}^{M} k_j \mathbf{v}_j$.

To efficiently handle with negative populations, the modified binomial $\tau$-leaping method was develop by (Chatterjee, Mayawala, Edwards, & Vlachos, 2005; Chatterjee, Vlachos, & Katsoulakis, 2005; Tian & Burrage, 2004). Furthermore, a number of variants and extensions of $\tau$-leaping have been developed to improve the performances as the implicit $\tau$

-leaping (Ahn, Han, & Sandu, 2015; Ahn & Sandu, 2011a, 2011b; Cao, Gillespie, & Petzold, 2007; Rathinam, Petzold, Cao, & Gillespie, 2003; Sandmann, 2009), the $k_a$ leaping (Peng & Wang, 2007) and the K-leaping and R-leaping (Auger, Chatelain, & Koumoutsakos, 2006; Cai & Xu, 2007).

### 2.2.2 Chemical Langevin method

The chemical Langevin method (Gillespie, 2000, 2001, 2002) further approximates the $\tau$-leaping method. Here, the value of $\tau$ must satisfy both the leap condition and the so-called *chemical Langevin condition* , that is, $a_j(\mathbf{x})\tau \gg 1$ for all reactions $R_j$. Under these conditions, the Poisson distribution $Poi(a(\mathbf{x})\tau)$ can be approximate to a Normal distribution

$$Poi\left(a_j(\mathbf{x})\tau\right) \approx N\left(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau\right) = a_j(\mathbf{x})\tau + \sqrt{a_j(\mathbf{x})\tau} N(0,1). \tag{8}$$

Therefore, the state is updated as

$$X(t+\tau) = \mathbf{x} + \sum_{j=1}^{M} a_j(\mathbf{x})\mathbf{v}_j\tau + \sum_{j=1}^{M} \sqrt{a_j(\mathbf{x})\tau} N(0,1)\mathbf{v}_j \tag{9}$$

that is equivalent to a stochastic differential equation (SDE) called chemical Langevin equation (CLE). Indeed, with the CLE, we approximate the stochastic discrete dynamics with a stochastic continuous dynamics by moving from a Poisson to a Gaussian probability distribution. As proved in the mathematical dissertations in Kurtz (1976, 1978), the CLE may also be derived by comparing a diffusion process with a Poisson process, independently from the $\tau$-Leaping method.

Algorithm 5 provides the pseudocode of the chemical Langevin method. For each step, a $\tau$ that satisfies the leap and the chemical Langevin conditions is selected (Step 4). Then, $M$ unit normal random numbers $n_j$ for $j = 1, \ldots, M$ are generated (Step 5) and are used to update the state (Step 6) according to Equation 9.

**Algorithm 5** Chemical Langevin Method

**Input:** a biochemical reaction network of $M$ reactions in which each reaction $R_j, j = 1, \ldots, M$, has a state change vector $\mathbf{v}_j$ and a propensity formula $a_j$, the initial state $\mathbf{x}_0$ at time 0, the simulation ending time $T_{max}$ and the error control parameter $\varepsilon$.

**Output:** a trajectory $X(t)$, $0 \le t \le T_{max}$, of the biochemical reaction network.

1: initialize the time $t = 0$ and the state $X = \mathbf{x}_0$.

2: **while** $(t < T_{max})$ **do.**

3: compute $a_j$ for each reaction $R_j$ with $j = 1, \ldots, M$.

4: determine $\tau$ that satisfies the leap condition and $a_j \tau \gg 1, \ \forall j = 1, \ldots, M$.

5: generate $M$ random number $n_j$ in $N(0, 1)$.

6: update $X = X + \sum_{j=1}^{M} a_j \tau \mathbf{v}_j + \sum_{j=1}^{M} n_j \mathbf{v}_j \sqrt{a_j \tau}$.

7: set $t = t + \tau$.

8: **end while.**

The simulation approach of the CLE has been further studied and improved with adaptive time stepping strategies (Ilie, 2012; Ilie & Teslya,

2012) and a method for approximating the CLE solution ("Automatic simulation of the chemical Langevin equation," Ilie & Morshed, 2013).

The chemical Langevin method directly links the stochastic simulation and the deterministic algorithms as described in Gillespie (2001, 2009). Consider Equation 9, in the limit case $a_j(\mathbf{x})\tau \to \infty$, the last term of the equation becomes negligibly small with respect to the second term and can be ignored, allowing the use of a deterministic approach to simulate the model. It is important to note that this is not the only approach aimed at deterministically describing a system that is intrinsically stochastic. Among the considerable body of literature, we refer to the works (Kurtz, 1976) and (Kurtz, 1978), in which rigorous mathematical arguments are provided to move from a stochastic to a deterministic description by dividing Equation 9 by the system volume.

The deterministic approach neglects either intrinsic or extrinsic noises and, once the initial conditions are defined, each run of the simulation returns the same dynamics that is close to the average of an hypothetically infinite number of stochastic simulations starting from the same initial state. This approximation usually counterbalances the loss in accuracy with a speed up of the simulation run time, but as exemplified in the introduction, it is not always the case. Since in this work we focus on stochastic algorithms, we refer to Burden, Faires, and Burden, (2016); Butcher (2008); Marchetti et al. (2017); Press, Teukolsky, Vetterling, and Flannery (2007); and Quarteroni, Sacco, and Salieri (2007) for the details about the deterministic approach.

## 2.3 Hybrid methods

Hybrid simulation methods combine and take advantage of both exact and approximate simulation algorithms. The idea behind the hybrid approach is to partition the system according to some features of each model reaction, and to apply different ad hoc simulation methods that

better fit the different partition features. The hybrid approach must face two main challenges, that is, to properly partition the model reactions and to synchronize these partitions after a simulation step.

The partition of the system aims at identifying the main sources of stochasticity of the system, in order to preserve the main sources and remove the negligible ones. Usually, the system partitioning is performed through ad hoc threshold-based algorithms, such as the reaction-based system partitioning (Griffith, Courtney, Peccoud, & Sanders, 2006; Irizarry, 2011; Salis & Kaznessis, 2005), which splits the system into fast and slow model reactions. Fast and slow reactions can be defined by considering the abundances of system species (Neogi, 2004), the propensities of the reactions (Haseltine & Rawlings, 2002), or a combination of the two (Alfonsi, Cancès, Turinici, Di Ventura, & Huisinga, 2005; Salis & Kaznessis, 2005). In general, the fast reactions involve frequent events or high-abundant species and, on the contrary, the slow reaction includes rare events or low-abundant species. Another important point is that the partitioning may depend on the system state that evolves over time. Therefore, the use of dynamical partitioning algorithms (Hepp et al., 2015; Marchetti, Priami, & Thanh, 2016; Neogi, 2004), which update the system division as the system evolves, can improve the accuracy of the simulation. After the reaction partitioning, different simulation algorithms are applied to each reaction subsystem and they are then synchronized (see Figure 3). Between the firing of two slow reactions, the set of fast reactions can evolve independently on the slow ones and its behavior can be simulated according to a deterministic or an approximate stochastic method. On the contrary, slow reactions require in general exact stochastic simulation. Since the reaction propensities $a_j$ of slow reactions often depend on species that are transformed by fast reactions, the two simulation approaches require continuous synchronization to guarantee that the simulation accuracy of slow reactions is not affected by the approximate simulation of fast reactions.

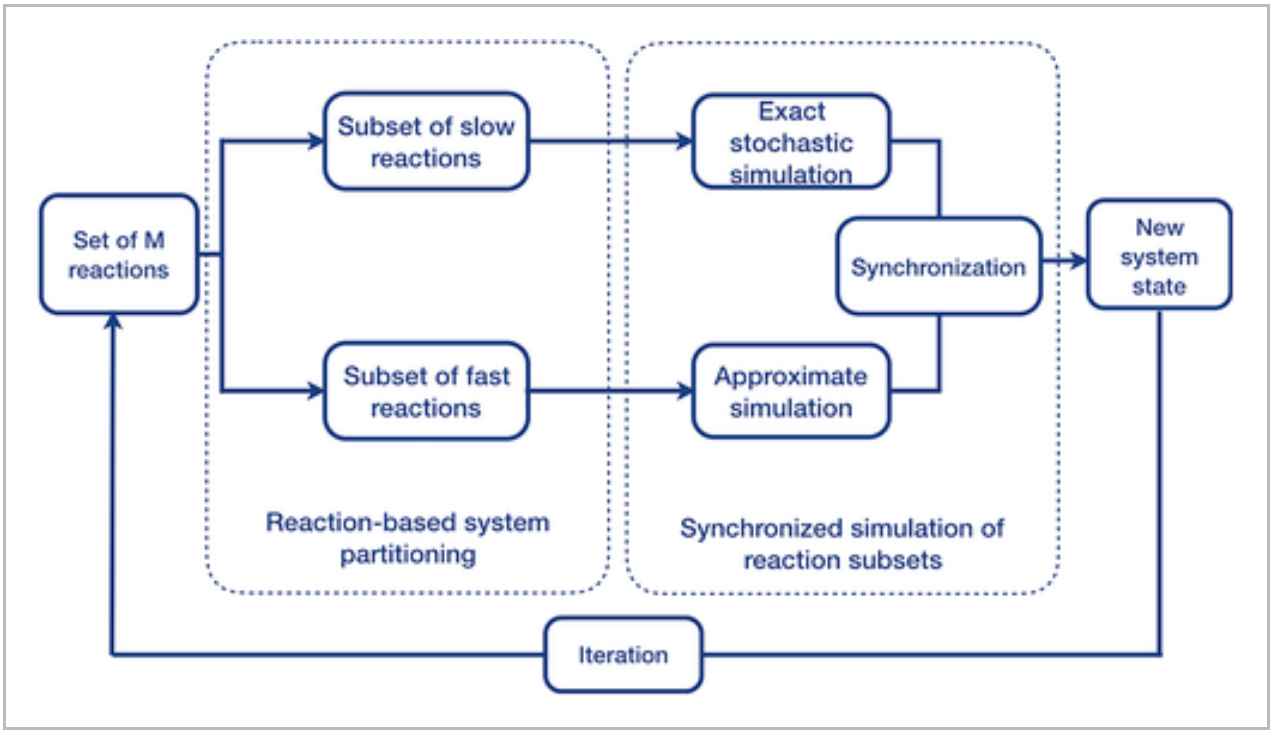



**Figure 3**

Open in figure viewer | PowerPoint

Schematic overview of a hybrid simulation strategy in which the set of reactions is divided into two subsets of slow and fast reactions. The two subsets are simulated with different methods and then are synchronized in order to define the new system state

To preserve the exactness of the simulation of slow reactions and to account for the changes in slow reaction propensities due to the simulation of fast reactions, the pdf (Equation 2) has to be varied to account for *time-varying transition propensities* . In this case, the pdf is defined as

$$p^s(\tau,\mu \mid \mathbf{x},t) = a_\mu(X(t+\tau))e^{-\int_t^{t+\tau} a_0^s(X(t'))\mathrm{d}t'}, \tag{10}$$

where $X(t+\tau)$ is the system state at time $t+\tau$, $\mathbf{x}$ is the current system

state and $a_0^s$ is the sum of the propensities of slow reactions. The firing time $\tau$ of the next slow reaction $R_\mu$ can be computed as the zero of the following equation (Marchetti et al., 2016; Pahle, 2008)

$$\int_t^{t+\tau} a_0^s(X(t'))\,\mathrm{d}t' + \ln(r) = 0. \tag{11}$$

Equation 11 is computationally challenging to solve because fast reactions change the system state during the time interval of $[t, t+\tau]$. Consequently, the hybrid simulation must simulate the fast reactions and simultaneously solve the equation to correctly generate the next slow reaction event. To overcome this computational effort, some strategies compute $\tau$ by solving an approximated equation that relax the constraint of Equation 11. We refer to Marchetti et al. (2017) for a more detailed discussion on partitioning and synchronization problems. In the literature, several hybrid algorithms have been introduced (Bentele & Eils, 2005; Griffith et al., 2006; Haseltine & Rawlings, 2002; Neogi, 2004) that differ in the reaction partitioning procedure and the selected stochastic algorithm. In the following, we will go into the details of the HRSSA, which relies on the concept of fluctuation interval of the system state introduced for RSSA in Section 2.1. This methods has the significant advantage that the use of propensity bounds allows the exact simulation of slow reactions without computing the integral of Equation 11, thus improving both the simulation accuracy and runtime.

### 2.3.1 Hybrid rejection-based SSA

The HRSSA (Marchetti et al., 2016) is an hybrid algorithm that is built on the top of the rejection-based SSA (Section 2.1). Like RSSA, it relies on the concept of fluctuation interval that allows the algorithm to work with

propensity bounds, rather than with exact propensities, to efficiently perform both the reaction partitioning and to manage with the time-varying transition propensities. HRSSA adopts a reaction-based partitioning algorithm that divides the system reactions into two subsystems of slow and fast reactions, $\mathscr{R}^s$ and $\mathscr{R}^f$, respectively. The partitioning of the system is updated only when the current system state exits the fluctuation interval. In this way, HRSSA reduces the runtime by decreasing the number of updates of reaction partitioning without losing the accuracy of the classification. In Algorithm 6, we describe the two class partitioning algorithm that HRSSA uses to classify a reaction $R_j$ in the fast or in the slow sets.

**Algorithm 6** Two class reaction-based partitioning

**Input:** a biochemical reaction system with stoichiometric matrix **v** and the lower bound of the system state $\underline{\mathbf{X}}$, the time increment $\tau^f$ for the approximate simulation of fast reactions, the parameter $\theta$ defining the minimum number of times that a fast reaction must be applied, on average, within the time range $\tau^f$ and the parameter $\gamma$ indicating how fine grained the reactant and product species must be in order to be approximated by continuous numbers.

**Output:** the sets $\mathscr{R}^s$ and $\mathscr{R}^f$ providing slow and fast reactions.

1: initialize $\mathscr{R}^s \varnothing$ and $\mathscr{R}^f = \varnothing$.

2: **for all** reactions $R_j \in \mathscr{R}$ **do.**

3: **if** ($a_j(\underline{\mathbf{x}})\tau^f < \theta$) **then.**

4: add reaction $R_j$ to the set $\mathscr{R}^s$.

5: **else if** (∃ species $S_i$ reactant or product of $R_j$ s.t. $\underline{x}_i \leq \gamma \cdot \mathbf{v}_{ji}$) **then.**

6: add reaction $R_j$ to the set $\mathscr{R}^s$.

7: **else.**

8: add reaction $R_j$ to the set $\mathscr{R}^f$.

9: **end if.**

10: **end for**

Slow reactions must satisfy at least one of the two following constraints:

$$a_j(\underline{\mathbf{x}})\tau^f < \theta,$$

(12)

$$\exists S_i, \text{ modified by } R_j, \text{ such that } \underline{x}_i < \gamma \cdot \mathbf{v}_{ji}.$$

(13)

These two conditions partition the model reactions on the basis of the frequency of reaction events and on the effect of each reaction on the species of the system. The key idea underlying the definition of these constraints is to identity the slow reactions as those representing (1) rare stochastic events and/or (2) affecting the dynamics of low numbered chemical species, which are those more influenced by the stochasticity. The first constraint requires that a slow reaction has a low average probability to occur during the simulation time interval of fast reactions. The second constraint requires the existence of at least one low-abundant species that is consistently affected by the occurrence of reaction $R_j$. The use of lower bounds for the reaction partitioning calculation does not affect the accuracy. Conversely, the use of lower bounds imposes tighter constraints that increase the number of reactions classified as slow.

After the reaction partitioning, HRSSA synchronizes the simulation of slow and fast reactions by considering the sum of the upper propensity bounds $a_0^s(\overline{\mathbf{x}})$ of the slow reactions, in spite of computing the integral of Equation 11. Then, the firing time of a candidate slow reaction is computed as

$$\tau = -\ln(r)/a_0^s(\overline{\mathbf{x}}),$$

(14)

where $r$ is a uniform random number in the interval (0, 1). Under the hypothesis that the system state remains inside its fluctuation interval during the time interval $[t, t + \tau]$, we can consider $a_0^s(\overline{\mathbf{x}})$ as a constant

independent from the time. This allows the algorithm to ignore the time-varying transition propensities and simulate the fast reactions over that time interval without taking into account of any effect on the slow reaction simulation.

**Algorithm 7** Hybrid Rejection-Based SSA

**Input:** a biochemical reaction system with initial state $X_0$, the parameter $\delta$ for calculating the fluctuation interval of the system state, the parameters $\tau^f$, $\theta$, and $\gamma$ for running Algorithm 6 and the simulation ending time $T_{max}$.

**Output:** a trajectory of the biochemical system.

1: initialize the time $t = 0$ and the state $X = X_0$.

2: **while** ($t < T_{max}$) **do.**

3: compute the fluctuation interval $\left[\underline{X}, \overline{X}\right] = \mathbb{X}$.

4: **for all** reactions $R_j \in \mathscr{R}$ **do.**

5: compute the reaction propensity bounds $\overline{a}_j$ and $\overline{a}_j$.

6: update the reaction partitioning (sets $\mathscr{R}^s$ and $\mathscr{R}^f$) by applying Algorithm 6 according to the input parameters $\gamma$, $\theta$ and $\tau^f$.

7: **end for.**

8: compute $\overline{a}_0^s = \sum_{R_j \in \mathcal{R}^s} \overline{a}_j$.

9: set *updateNeeded* = *false* .

10: **while** ($t < T_{max} \wedge \neg$ *updateNeeded* ) **do.**

Algorithm 7 provides a HRSSA implementation. Briefly, after the reaction

partitioning, HRSSA proceeds as follow:

1. Compute the total upper propensity bound $\overline{a_0}^s = \sum_{R_j \in \mathcal{R}^s} \overline{a}_j = \sum_{R_j \in \mathcal{R}^s} a_j(\overline{\mathbf{x}})$ for the reactions in $\mathcal{R}^S$ (Step 8).
2. Compute the firing time $\tau$ of the next slow reaction (Step 11).
3. Simulate the fast reactions over the time interval $\tau'$ independently from the slow reactions, where $\tau'$ is the largest time step smaller than $\tau$, such that the system state remains inside its fluctuation interval (Step 12).
4. If it is possible to simulate fast reactions until $\tau$ without violating the fluctuation interval ($\tau' = \tau$), select a reaction $R_\mu$ in $\mathcal{R}^S$ and validate its firing according to the RSSA rejection test (steps 14–15);
5. If $\tau' < \tau$, the simulation of fast reactions violates the constraint on the fluctuations interval and all the simulation of the fast reactions must be discarded (Step 21). This step, which imposes a constraint that is more stringent with respect to the one presented in the original HRSSA paper (Marchetti et al., 2016), has been introduced to guarantee the exactness of the simulation of slow reactions.
6. In all cases, when the system state exists the fluctuation interval, both the fluctuation interval and the reaction partitioning are updated (steps 5–6).

# 3 A SIMULATION TEST-CASE IN COMPUTATIONAL SYSTEM BIOLOGY: THE SPHINGOLIPID METABOLISM

In this section, we exemplify how different simulation algorithms may unveil different insights into biological systems by providing different levels of accuracy. To this end, we consider two models that have been developed in the context of system biology to study the metabolism of a family of lipid molecules, called sphingolipids and, in particular, the

ceramide. These models were simulated using both deterministic and stochastic approaches. Here, rather than focusing on the performance of the algorithms, we show how the use of two exact SSAs, RSSA, and its variant that accounts for extrinsic noise RSSA (Thanh et al., 2018), may provide a more reliable simulation of a system dynamics, especially when low-abundance species are involved. Moreover, incorporating the extrinsic noise reduces the approximations that are introduced in the modeling process and further improves the reliability of the *in silico* results. For an accurate benchmarking of all the reviewed simulation algorithms, we refer to Marchetti et al. (2017).

Ceramide and the sphingolipids are bioactive lipids that play crucial roles in many aspects of cell fate and regulations (Merrill, 2011). For example, sphingolipids are fundamental components of the plasma membrane, and, at the same time, they play a role in different signaling pathways, as well as in the development of several diseases. As a consequence, sphingolipids affect different cell regulatory processes, such as inflammation, proliferation, apoptosis, and energy metabolism. For example, the accumulation of ceramide (Cer) decreases the cell glucose uptake and promotes the insulin resistance, ceramide-1-phosphate (Cer1P) acts as a secondary messenger that contributes to the regulation of inflammation and cell survival. Analogously, sphingosine-1-phosphate (S1P) dysregulation affects cell survival and, in addition, it contributes to the development of diabetes and tumor growth. Notably, the accumulation of certain sphingolipids is the base of certain rare genetic diseases, such as Niemann-Pick and Gaucher diseases (Platt, 2014).

During the last two decades, different models have been proposed to mechanistically explore different aspects of the sphingolipid metabolism (Alvarez-Vasquez et al., 2005; Gupta et al., 2011; Kaddi, Niesner, et al., 2018; Kaddi, Reali et al., 2018; Reali et al., 2017; Wronowska, Charzyńska, Nienałtowski, & Gambin, 2015). Here, we analyze two models that have

been simulated using three different approaches, and each approach has led to different discoveries.

The first model we consider was developed in 2011 by Gupta et al. (2011) to provide a mechanistic description of the *de novo* synthesis of ceramide production and the sphingomyelinase pathway. The authors developed a model that connects the two biological layers of gene expression and lipidomics data, in order to simulate two experimental conditions, that is, control and $Kdo_2$-Lipid-A-induced inflammation. The authors used a deterministic approach, in particular, a Runge–Kutta method of orders 2 and 3 (Marchetti et al., 2017; Quarteroni et al., 2007) that permits a fast simulation of the model. The use of a deterministic method allowed the use of computationally consuming procedures that require many simulations of the model, such as parameter estimation techniques to determine unknown rates, and deterministic parametric sensitivity analysis to assess the robustness of the model.

In 2017, Reali, Morine, et al. (2017) have extended the Gupta et al. model to link the ceramide metabolism with the sphingolipids *salvage* pathway. This model is used in combination with gene expression data from independent experiments and sensitivity analysis to explore the effect of the accumulation of Cer, S1P, and glucosylceramide (GluCer) on the development of insulin resistance. More in detail, the model is simulated using both a deterministic approach and an exact SSA with different ends. The deterministic simulation, also in this case a Runge–Kutta approach, allowed the use of computationally intensive procedures. On the other hand, the more accurate RSSA allowed at quantifying the effect of stochastic noise on the model results. The comparison of the exact simulation and the approximate deterministic simulation results in Figure 4 clearly suggests that stochastic noise may play a relevant role in determining the behavior of the less abundant species, such as Cer1P and S1P. In addition, the stochastic simulation coupled with the use of specific

scaling factors has provided useful information on the scale at which the model may provide reliable insights. For example, their results show that the deterministic results for S1P are not reliable when a scaling factor of 1e-03 is applied to reduce the population abundance (Figure 4). This scaling factor is consistent with considering a reduced volume, as is the case of considering an intracellular compartment. We refer to the supplementary material in Reali, Morine, et al. (2017) for the analysis of the effect of introducing this scaling factor in the model. In this reduced volume condition, the comparison between the two methods have highlighted how the stochastic noise plays a major role at that scale and exact or hybrid SSAs should be preferred to the deterministic simulations.

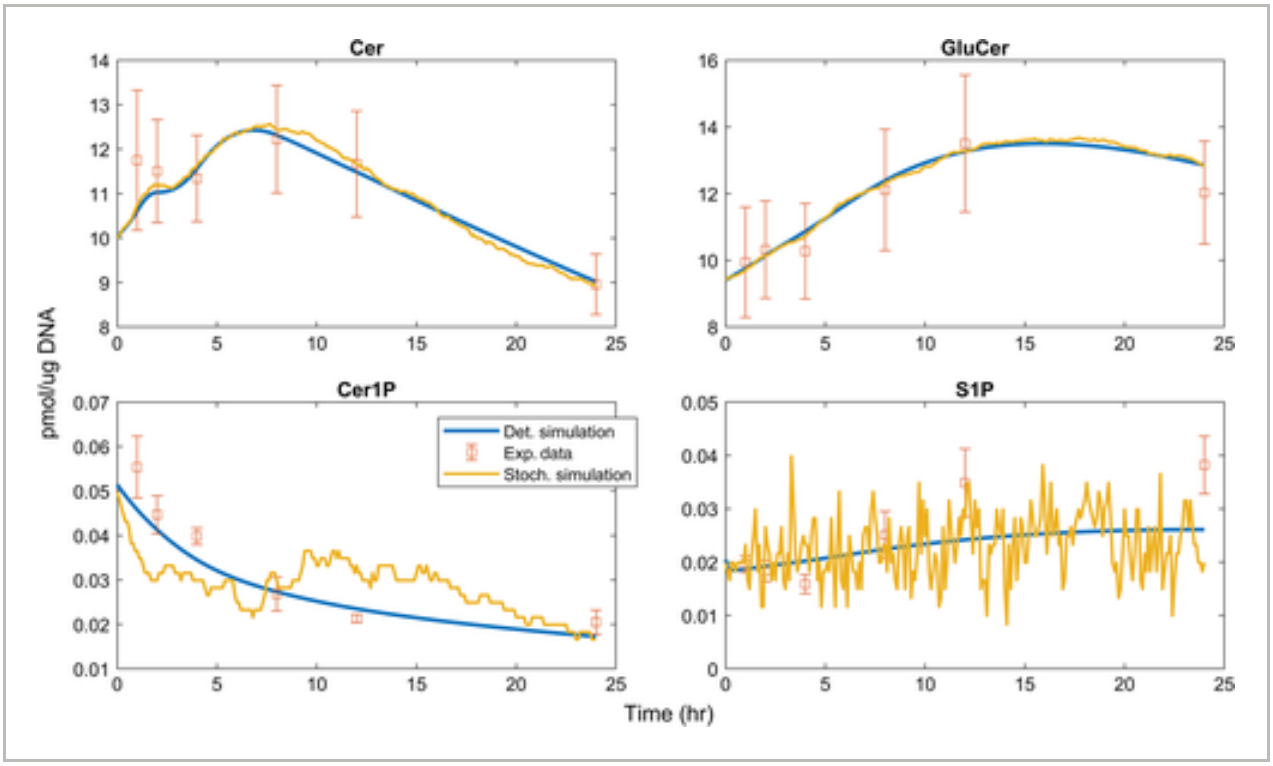



**Figure 4**

Open in figure viewer | PowerPoint

The comparison of the simulation results of Reali et al. model using a deterministic approach and rejection-based SSA (RSSA) in combination with a scaling factor of 1e-03 to reduce the population abundance. The results highlight how the stochastic noise is affecting the less abundant

species ceramide-1-phosphate (Cer1P) and sphingosine-1-phosphate (S1P), suggesting the use of exact or hybrid stochastic simulation algorithms

In 2018, Thanh et al. (2018) have extended RSSA to account for both intrinsic and extrinsic noise and, in their work, the authors demonstrated how the combination of these noises may play a significant role in the simulation of Reali et al. model. In fact, Thanh et al. analyzed the model results and the method performance, and showed that by accounting for both the noises, the results for some sphingolipids, such as dihydrosphingosine (dhSph), Cer1P, and S1P may vary significantly at certain time points. Figure 5 depicts the probability distribution of the number of molecules for Cer1P and S1P at the end of the simulation ($t$ = 24 hr) and shows how the more accurate noise RSSA simulation differ from RSSA. Figure 5 exemplifies how more accurate simulation algorithms may better mimic the behavior of the sphingolipid metabolism. As a consequence of the importance of these sphingolipids, correctly describing their dynamics is crucial not only to accurately reproduce their behavior but also for improving our understanding of their effect on cell fate.

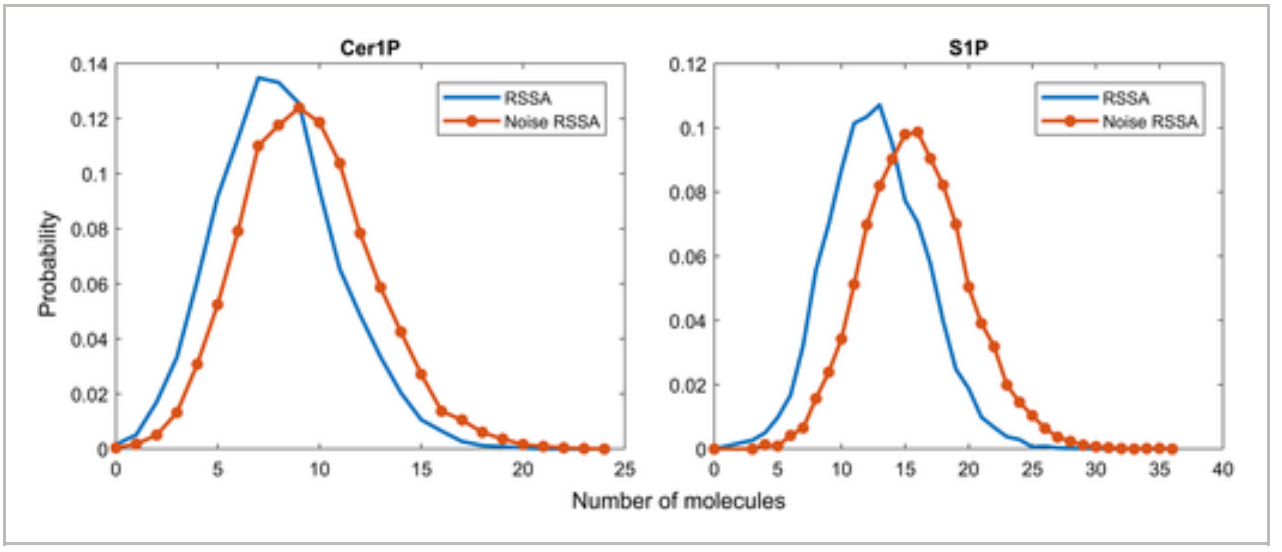



**Figure 5**

Open in figure viewer | PowerPoint

Distribution of the number of molecules for the species ceramide-1-phosphate (Cer1P) and sphingosine-1-phosphate (S1P) at the end of the simulation ($t$ = 24 hr) using rejection-based SSA (RSSA) and the RSSA version that accounts for extrinsic noise (10,000 simulation runs)

# 4 CONCLUSIONS

This review provides an organized summary of the main simulation approaches for computational systems biology. Starting from the formal description of a biochemical system as a system of reactions, we define a SSA as an event-driven simulation approach that is able to produce an accurate realization of the system state. During the last 50 years, several implementations of SSA have been developed that differ in the level of approximations introduced to simulate the system, and they can be divided into: exact, approximate, and hybrid methods. For each one of these strategies, we give an accurate description of what we consider the main algorithms in terms of historical importance and current use. In addition, the study of the sphingolipid models exemplifies the importance of how the use of different simulation algorithms may provide different accuracy and insights, and in the end, this computation tool may support a better understanding of biological problems.

# CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

# RELATED WIREs ARTICLES

Modeling cellular compartmentation in one-carbon metabolism

Rule-

based modeling: a computational approach for studying biomolecular site dynamics in cell signaling systems

Mathematical models in biology: from molecules to life

Kinetic models of hematopoietic differentiation

Stochastic approaches in systems biology

Ahn, T.-H., Han, X., & Sandu, A. (2015). Implicit simulation methods for stochastic chemical kinetics. *Journal of Applied Analysis and Computation*, **5**(3), 420– 452. https://doi-org.exlibris.colgate.edu/10.11948/2015034
Web of Science® | Google Scholar

Ahn, T.-H., & Sandu, A. (2011a). Fully implicit tau-leaping methods for the stochastic simulation of chemical kinetics. In *Proceedings of the 19th high performane computing symposia* (pp. 118–125), Boston, MA.
Google Scholar

Ahn, T.-H., & Sandu, A. (2011b). Implicit second order weak Taylor tau-leaping methods for the stochastic simulation of chemical kinetics. *Procedia Computer Science*, **4**, 2297– 2306.
https://doi-org.exlibris.colgate.edu/10.1016/j.procs.2011.04.250
Crossref | Google Scholar

Alfonsi, A., Cancès, E., Turinici, G., Di Ventura, B., & Huisinga, W. (2005). Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems. *ESAIM: Proceedings*, **14**, 1– 13.
https://doi-org.exlibris.colgate.edu/10.1051/proc:2005001
Crossref | Google Scholar

Alvarez-Vasquez, F., Sims, K. J., Cowart, L. A., Okamoto, Y., Voit, E. O., & Hannun, Y. A. (2005). Simulation and validation of modelled sphingolipid metabolism in *Saccharomyces cerevisiae* . *Nature*, **433**, 425– 430.

https://doi-org.exlibris.colgate.edu/10.1038/nature03232
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Anderson, D. F. (2007). A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *The Journal of Chemical Physics*, **127**(21), 214107.
https://doi-org.exlibris.colgate.edu/10.1063/1.2799998
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Anderson, D. F. (2008). Incorporating postleap checks in tau-leaping. *The Journal of Chemical Physics*, **128**(5), 054103.
https://doi-org.exlibris.colgate.edu/10.1063/1.2819665
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Anderson, D. F., Higham, D. J., & Sun, Y. (2016). Multilevel Monte Carlo for stochastic differential equations with additive small noise. *SIAM Journal on Numerical Analysis*, **54**(2), 505– 529.
https://doi-org.exlibris.colgate.edu/10.1007/s10479-009-0663-8
Crossref | Web of Science® | Google Scholar

Auger, A., Chatelain, P., & Koumoutsakos, P. (2006). R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps. *The Journal of Chemical Physics*, **125**(8), 084103.
https://doi-org.exlibris.colgate.edu/10.1063/1.2218339
Crossref | PubMed | Web of Science® | Google Scholar

Baker, R. E., Yates, C. A., & Erban, R. (2010). From microscopic to macroscopic descriptions of cell migration on growing domains. *Bulletin of Mathematical Biology*, **72**(3), 719– 762.
Crossref | PubMed | Web of Science® | Google Scholar

Bentele, M., & Eils, R. (2005). General stochastic hybrid method for the simulation of chemical reaction processes in cells. In *Computational methods in systems biology* (pp. 248–251).
Google Scholar

Burden, R. L., Faires, D. J., & Burden, A. M. (2016). *Numerical analysis* ( 10th ed.). Boston, MA: Cengage Learning.
Google Scholar

Butcher, J. C. (2008). *Numerical methods for ordinary differential equations* ( 2nd ed.). Hoboken, NJ: Wiley.
Wiley Online Library | Google Scholar

Cai, X., & Wang, X. (2007). Stochastic modeling and simulation of gene networks-a review of the state-of-the-art research on stochastic simulations. *IEEE Signal Processing Magazine*, **24**(1), 27– 36.
Crossref | CAS | Web of Science® | Google Scholar

Cai, X., & Xu, Z. (2007). K-leap method for accelerating stochastic simulation of coupled chemical reactions. *The Journal of Chemical Physics*, **126**(7), 074102. https://doi-org.exlibris.colgate.edu/10.1063/1.2436869
Crossref | PubMed | Web of Science® | Google Scholar

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2005). Avoiding negative populations in explicit Poisson tau-leaping. *The Journal of Chemical Physics*, **123**(5), 054104. https://doi-org.exlibris.colgate.edu/10.1063/1.1992473
Crossref | PubMed | Web of Science® | Google Scholar

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2006). Efficient step size selection for the tau-leaping simulation method. *The Journal of Chemical Physics*, **124**(5), 044109. https://doi-org.exlibris.colgate.edu/10.1063/1.2159468
Crossref | PubMed | Web of Science® | Google Scholar

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2007). Adaptive explicit-implicit tau-leaping method with automatic tau selection. *The Journal of Chemical Physics*, **126**(22), 224101. https://doi-org.exlibris.colgate.edu/10.1063/1.2745299
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Cao, Y., Li, H., & Petzold, L. R. (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The Journal of Chemical*

*Physics*, **121**(9), 4059– 4067.
https://doi-org.exlibris.colgate.edu/10.1063/1.1778376
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Cao, Y., & Petzold, L. R. (2005). Trapezoidal tau-leaping formula for the stochastic simulation of biochemical systems. In *Proceedings of foundations of systems biology in engineering (FOSBE 2005)* (pp. 149–152), Santa Barbara, CA.
Google Scholar

Chatterjee, A., Mayawala, K., Edwards, J. S., & Vlachos, D. G. (2005). Time accelerated Monte Carlo simulations of biological networks using the binomial $\tau$-leap method. *Bioinformatics*, **21**(9), 2136– 2137.
https://doi-org.exlibris.colgate.edu/10.1093/bioinformatics/bti308
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Chatterjee, A., Vlachos, D. G., & Katsoulakis, M. A. (2005). Binomial distribution based $\tau$-leap accelerated stochastic simulation. *The Journal of Chemical Physics*, **122**(2), 024412.
https://doi-org.exlibris.colgate.edu/10.1063/1.1833357
Crossref | Web of Science® | Google Scholar

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.
https://doi-org.exlibris.colgate.edu/10.1126/science.1070919
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Ethier, S. N., & Kurtz, T. G. (1986). Markov processes: Characterization and. *Convergence*, **43**, 484. https://doi-org.exlibris.colgate.edu/10.2307/2531839
Google Scholar

Fisher, J., & Henzinger, T. A. (2007). Executable cell biology. *Nature Biotechnology*, **25**(11), 1239– 1249.
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Gibson, M. A., & Bruck, J. (2000). Efficient exact stochastic simulation of

chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, **104**(9), 1876– 1889.
https://doi-org.exlibris.colgate.edu/10.1021/jp993732q
Crossref | CAS | Web of Science® | Google Scholar

Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, **22**(4), 403– 434.
https://doi-org.exlibris.colgate.edu/10.1016/0021-9991(76)90041-3
Crossref | CAS | Web of Science® | Google Scholar

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, **81**(25), 2340– 2361.
Crossref | CAS | Web of Science® | Google Scholar

Gillespie, D. T. (2000). The chemical Langevin equation. *The Journal of Chemical Physics*, **113**(1), 297– 306.
https://doi-org.exlibris.colgate.edu/10.1063/1.481811
Crossref | CAS | Web of Science® | Google Scholar

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, **115**(4), 1716– 1733. https://doi-org.exlibris.colgate.edu/10.1063/1.1378322
Crossref | CAS | Web of Science® | Google Scholar

Gillespie, D. T. (2002). The chemical Langevin and Fokker-Planck equations for the reversible isomerization reaction. *The Journal of Physical Chemistry A*, **106**(20), 5063– 5071. https://doi-org.exlibris.colgate.edu/10.1021/jp0128832
Crossref | CAS | Web of Science® | Google Scholar

Gillespie, D. T. (2009). Deterministic limit of stochastic chemical kinetics. *The Journal of Physical Chemistry B*, **113**(6), 1640– 1644.
https://doi-org.exlibris.colgate.edu/10.1021/jp806431b
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Gillespie, D. T., & Petzold, L. R. (2003). Improved lead-size selection for accelerated stochastic simulation. *The Journal of Chemical Physics*, **119**(16), 8229– 8234. https://doi-org.exlibris.colgate.edu/10.1063/1.1613254
Crossref | CAS | Web of Science® | Google Scholar

Griffith, M., Courtney, T., Peccoud, J., & Sanders, W. H. (2006). Dynamic partitioning for hybrid simulation of the bistable HIV-1 transactivation network. *Bioinformatics*, **22**(22), 2782– 2789. https://doi-org.exlibris.colgate.edu/10.1093/bioinformatics/btl465
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Gupta, S., Maurya, M. R., Merrill, A. H., Glass, C. K., & Subramaniam, S. (2011). Integration of lipidomics and transcriptomics data towards a systems biology model of sphingolipid metabolism. *BMC Systems Biology*, **5**(1), 26. https://doi-org.exlibris.colgate.edu/10.1186/1752-0509-5-26
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Halasz, A., Kumar, V., Imieliński, M., Belta, C., Sokolsky, O., Pathak, S., & Rubin, H. (2007). Analysis of lactose metabolism in *E. coli* using reachability analysis of hybrid systems. *IET Systems Biology*, **1**(2), 130– 148.
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Harris, L. A., & Clancy, P. (2006). A partitioned leaping approach for multiscale modeling of chemical reaction dynamics. *The Journal of Chemical Physics*, **125**(14), 144107. https://doi-org.exlibris.colgate.edu/10.1063/1.2354085
Crossref | PubMed | Web of Science® | Google Scholar

Haseltine, E. L., & Rawlings, J. B. (2002). Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of Chemical Physics*, **117**(15), 6959– 6969. https://doi-org.exlibris.colgate.edu/10.1063/1.1505860
Crossref | CAS | Web of Science® | Google Scholar

Hepp, B., Gupta, A., & Khammash, M. (2015). Adaptive hybrid simulations for multiscale stochastic reaction networks. *The Journal of Chemical Physics*,

**142**(3), 034118. https://doi-org.exlibris.colgate.edu/10.1063/1.4905196
Crossref | PubMed | Web of Science® | Google Scholar

Ilie, S. (2012). Variable time-stepping in the pathwise numerical solution of the chemical Langevin equation. *The Journal of Chemical Physics*, **137**(23), 234110. https://doi-org.exlibris.colgate.edu/10.1063/1.4771660
Crossref | PubMed | Web of Science® | Google Scholar

Ilie, S., & Teslya, A. (2012). An adaptive stepsize method for the chemical Langevin equation. *The Journal of Chemical Physics*, **136**(18), 184101. https://doi-org.exlibris.colgate.edu/10.1063/1.4711143
Crossref | PubMed | Web of Science® | Google Scholar

Ilie, S., & Morshed, M. (2013). Automatic simulation of the chemical Langevin equation. *Applied Mathematics*, **4**(1A), 235– 241. https://doi-org.exlibris.colgate.edu/10.4236/am.2013.41A036
Crossref | Google Scholar

Irizarry, R. (2011). Stochastic simulation of population balance models with disparate time scales: Hybrid strategies. *Chemical Engineering Science*, **66**(18), 4059– 4069. https://doi-org.exlibris.colgate.edu/10.1016/j.ces.2011.05.035
Crossref | CAS | Web of Science® | Google Scholar

Kaddi, C., Niesner, B., Baek, R., Jasper, P., Pappas, J., Tolsma, J., … Azer, K. (2018). Quantitative systems pharmacology modeling of acid sphingomyelinase deficiency and the enzyme replacement therapy olipudase alfa is an innovative tool for linking pathophysiology and pharmacology. *CPT Pharmacometrics & Systems Pharmacology*, **7**, 442– 452.
Wiley Online Library | CAS | PubMed | Web of Science® | Google Scholar

Kaddi, C., Reali, F., Marchetti, L., Niesner, B., Parolo, S., Simoni, G., … Azer, K. (2018). Integrated quantitative systems pharmacology (QSP) model of lysosomal diseases provides an innovative computational platform to support research and therapeutic development for the sphingolipidoses. *Molecular Genetics and Metabolism Systems Biology*, **123**, S73– S74.
Crossref | Web of Science® | Google Scholar

Kang, H.-W., & Kurtz, T. G. (2013). Separation of time-scales and model reduction for stochastic reaction networks. *The Annals of Applied Probability*, **23**(2), 529– 583. https://doi-org.exlibris.colgate.edu/10.1214/12-AAP841
Crossref | Web of Science® | Google Scholar

Kurtz, T. G. (1976). Limit theorems and diffusion approximations for density dependent markov chains. In R. J.-B. Wets (Ed.), *Stochastic systems: Modeling, identification and optimization, i* (pp. 67– 78). Berlin, Heidelberg: Springer. https://doi-org.exlibris.colgate.edu/10.1007/BFb0120765
Crossref | Google Scholar

Kurtz, T. G. (1978). Strong approximation theorems for density dependent Markov chains. *Stochastic Processes and their Applications*, **6**(3), 223– 240. https://doi-org.exlibris.colgate.edu/10.1016/0304-4149(78)90020-0
Crossref | Google Scholar

Li, H., & Petzold, L. R. (2006). Logarithmic direct method for discrete stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, **16**, 1– 11.
Google Scholar

Li, Y., & Hu, L. (2015). A fast exact simulation algorithm for a class of markov jump processes. *The Journal of Chemical Physics*, **143**(18), 184105. https://doi-org.exlibris.colgate.edu/10.1063/1.4934972
Crossref | PubMed | Web of Science® | Google Scholar

Lok, L., & Brent, R. (2005). Automatic generation of cellular reaction networks with Moleculizer 1.0. *Nature Biotechnology*, **23**(1), 131– 136. https://doi-org.exlibris.colgate.edu/10.1038/nbt1054
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Lotka, A. J. (1920). Analytical note on certain rhythmic relations in organic systems. *Proceedings of the National Academy of Sciences*, **6**, 410– 415.
Crossref | CAS | PubMed | Google Scholar

Marchetti, L., Priami, C., & Thanh, V. H. (2016). HRSSA—Efficient hybrid stochastic simulation for spatially homogeneous biochemical reaction networks. *Journal of Computational Physics*, **317**, 301– 317.
https://doi-org.exlibris.colgate.edu/10.1016/j.jcp.2016.04.056
Crossref | CAS | Web of Science® | Google Scholar

Marchetti, L., Priami, C., & Thanh, V. H. (2017). *Simulation algorithms for computational systems biology*. Cham, Switzerland: Springer-Verlag New York Inc.
Crossref | Google Scholar

Mauch, S., & Stalzer, M. (2011). Efficient formulations for exact stochastic simulation of chemical systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **8**(1), 27– 35.
https://doi-org.exlibris.colgate.edu/10.1109/TCBB.2009.47
Crossref | PubMed | Web of Science® | Google Scholar

McAdams, H. H., & Arkin, A. (1999). It's a noisy business! Genetic regulation at the nanomolar scale. *Trends in Genetics*, **15**(2), 65– 69.
https://doi-org.exlibris.colgate.edu/10.1016/S0168-9525(98)01659-X
Crossref | CAS | PubMed | Web of Science® | Google Scholar

McCollum, J. M., Peterson, G. D., Cox, C. D., Simpson, M. L., & Samatova, N. F. (2006). The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Computational Biology and Chemistry*, **30**(1), 39– 49.
https://doi-org.exlibris.colgate.edu/10.1016/j.compbiolchem.2005.10.007
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Merrill, A. H. (2011). Sphingolipid and glycosphingolipid metabolic pathways in the era of sphingolipidomics. *Chemical Reviews*, **111**(10), 6387– 6422.
https://doi-org.exlibris.colgate.edu/10.1021/cr2002917
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Misselbeck, K., Marchetti, L., Field, M. S., Scotti, M., Priami, C., & Stover, P. J. (2017). A hybrid stochastic model of folate-mediated one-carbon metabolism:

Effect of the common c677t mthfr variant on de novo thymidylate biosynthesis. *Scientific Reports*, **7**(1), 797.
Crossref | PubMed | Web of Science® | Google Scholar

Misselbeck, K., Marchetti, L., Priami, C., Stover, P. J., & Field, M. S. (2019). The 5-formyltetrahydrofolate futile cycle reduces pathway stochasticity in an extended hybrid-stochastic model of folate-mediated one-carbon metabolism. *Scientific Reports*, **9**(1), 4322.
Crossref | PubMed | Web of Science® | Google Scholar

Moraes, A., Tempone, R., & Vilanova, P. (2014). Hybrid chernoff tau-leap. *Multiscale Modeling and Simulation*, **12**(2), 581– 615.
Crossref | Web of Science® | Google Scholar

Moraes, A., Tempone, R., & Vilanova, P. (2015). Multilevel hybrid chernoff tau-leap. *BIT Numerical Mathematics*, **56**(1), 189– 239.
https://doi-org.exlibris.colgate.edu/10.1007/s10543-015-0556-y
Crossref | Web of Science® | Google Scholar

Neogi, N. A. (2004). *Dynamic partitioning of large discrete event biological systems for hybrid simulation and analysis*. Berlin and Heidelberg, Germany: Springer.
Crossref | Google Scholar

Pahle, J. (2008). Biochemical simulations: Stochastic, approximate stochastic and hybrid approaches. *Briefings in Bioinformatics*, **10**(1), 53– 64.
https://doi-org.exlibris.colgate.edu/10.1093/bib/bbn050
Crossref | Web of Science® | Google Scholar

Peng, X.-J., & Wang, Y.-F. (2007). L-leap: Accelerating the stochastic simulation of chemically reacting systems. *Applied Mathematics and Mechanics*, **28**(10), 1361– 1371. https://doi-org.exlibris.colgate.edu/10.1007/s10483-007-1009-y
Crossref | Google Scholar

Platt, F. M. (2014). Sphingolipid lysosomal storage disorders. *Nature*, **510**, 68–

75.
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge, England: Cambridge University Press.
Google Scholar

Priami, C., & Morine, M. J. (2015). *Analysis of biological systems*. London, England: Imperial College Press.
Crossref | Google Scholar

Quarteroni, A., Sacco, R., & Salieri, F. (2007). *Numerical mathematics*. New York, NY: Springer.
Crossref | Google Scholar

Ramaswamy, R., González-Segredo, N., & Sbalzarini, I. F. (2009). A new class of highly efficient exact stochastic simulation algorithms for chemical reaction networks. *The Journal of Chemical Physics*, **130**(24), 244104.
https://doi-org.exlibris.colgate.edu/10.1063/1.3154624
Crossref | PubMed | Web of Science® | Google Scholar

Ramaswamy, R., & Sbalzarini, I. F. (2010). A partial-propensity variant of the composition-rejection stochastic simulation algorithm for chemical reaction networks. *The Journal of Chemical Physics*, **132**(4), 044102.
https://doi-org.exlibris.colgate.edu/10.1063/1.3297948
Crossref | PubMed | Web of Science® | Google Scholar

Rathinam, M., Petzold, L. R., Cao, Y., & Gillespie, D. T. (2003). Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, **119**(24), 12784– 12794.
https://doi-org.exlibris.colgate.edu/10.1063/1.1627296
Crossref | CAS | Web of Science® | Google Scholar

Reali, F., Morine, M. J., Kahramanoggullari, O., Raichur, S., Schneider, H.-C.,

Crowther, D., & Priami, C. (2017). Mechanistic interplay between ceramide and insulin resistance. *Scientific Reports*, **7**, 41231.
Crossref | PubMed | Web of Science® | Google Scholar

Reali, F., Priami, C., & Marchetti, L. (2017). Optimization algorithms for computational systems biology. *Frontiers in Applied Mathematics and Statistics*, **3**, 6. http://doi.org.exlibris.colgate.edu:2048/10.3389/fams.2017.00006
Crossref | PubMed | Google Scholar

Salis, H., & Kaznessis, Y. (2005). Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *The Journal of Chemical Physics*, **122**(5), 054103.
https://doi-org.exlibris.colgate.edu/10.1063/1.1835951
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Sandmann, W. (2009). Streamlined formulation of adaptive explicit-implicit tau-leaping with automatic tau selection. In *Proceedings of winter simulation conference* (pp. 1104–1112).
https://doi-org.exlibris.colgate.edu/10.1109/WSC.2009.5429309
Google Scholar

Sanft, K. R., & Othmer, H. G. (2015). Constant-complexity stochastic simulation algorithm with optimal binning. *The Journal of Chemical Physics*, **143**(7), 074108. https://doi-org.exlibris.colgate.edu/10.1063/1.4928635
Crossref | PubMed | Web of Science® | Google Scholar

Santillán, M., & Mackey, M. C. (2008). Quantitative approaches to the study of bistability in the lac operon of *Escherichia coli* . *Journal of the Royal Society Interface*, **5**(suppl_1), S29– S39.
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Schulze, T. (2008). Efficient kinetic Monte Carlo simulation. *Journal of Computational Physics*, **227**(4), 2455– 2462.
https://doi-org.exlibris.colgate.edu/10.1016/j.jcp.2007.10.021
Crossref | Web of Science® | Google Scholar

Slepoy, A., Thompson, A. P., & Plimpton, S. J. (2008). A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *The Journal of Chemical Physics*, **128**(20), 205101. https://doi-org.exlibris.colgate.edu/10.1063/1.2919546
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Swain, P. S., Elowitz, M. B., & Siggia, E. D. (2002). Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proceedings of the National Academy of Sciences*, **99**(20), 12795– 12800. https://doi-org.exlibris.colgate.edu/10.1073/pnas.162041399
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Thanh, V. H., Marchetti, L., Reali, F., & Priami, C. (2018). Incorporating extrinsic noise into the stochastic simulation of biochemical reactions: A comparison of approaches. *The Journal of Chemical Physics*, **148**(6), 064111. https://doi-org.exlibris.colgate.edu/10.1063/1.5016338
Crossref | PubMed | Web of Science® | Google Scholar

Thanh, V. H., & Priami, C. (2015). Simulation of biochemical reactions with time-dependent rates by the rejection-based algorithm. *The Journal of Chemical Physics*, **143**(5), 054104. https://doi-org.exlibris.colgate.edu/10.1063/1.4927916
Crossref | PubMed | Web of Science® | Google Scholar

Thanh, V. H., Priami, C., & Zunino, R. (2014). Efficient rejection-based simulation of biochemical reactions with stochastic noise and delays. *The Journal of Chemical Physics*, **141**(13), 134116. https://doi-org.exlibris.colgate.edu/10.1063/1.4896985
Crossref | PubMed | Web of Science® | Google Scholar

Thanh, V. H., & Zunino, R. (2012). Tree-based search for stochastic simulation algorithm. In *Proceedings of the ACM symposium on applied computing* (pp. 1415–1416). https://doi-org.exlibris.colgate.edu/10.1145/2245276.2232001
Google Scholar

Thanh, V. H., & Zunino, R. (2014). Adaptive tree-based search for stochastic

simulation algorithm. *International Journal of Computational Biology and Drug Design*, **7**(4), 341– 357.
Crossref | CAS | PubMed | Google Scholar

Thanh, V. H., Zunino, R., & Priami, C. (2015). On the rejection-based algorithm for simulation and analysis of large-scale reaction networks. *The Journal of Chemical Physics*, **142**(24), 244106.
https://doi-org.exlibris.colgate.edu/10.1063/1.4922923
Crossref | PubMed | Web of Science® | Google Scholar

Thanh, V. H., Zunino, R., & Priami, C. (2017a). Efficient constant-time complexity algorithm for stochastic simulation of large reaction networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **14**(3), 657–667. https://doi-org.exlibris.colgate.edu/10.1109/TCBB.2016.2530066
Crossref | PubMed | Web of Science® | Google Scholar

Thanh, V. H., Zunino, R., & Priami, C. (2017b). Efficient stochastic simulation of biochemical reactions with noise and delays. *The Journal of Chemical Physics*, **146**(8), 084107. https://doi-org.exlibris.colgate.edu/10.1063/1.4976703
Crossref | PubMed | Web of Science® | Google Scholar

Tian, T., & Burrage, K. (2004). Binomial leap methods for simulating stochastic chemical kinetics. *The Journal of Chemical Physics*, **121**(21), 10356– 10364.
https://doi-org.exlibris.colgate.edu/10.1063/1.1810475
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. *Nature*, **118**, 558– 560.
Crossref | Web of Science® | Google Scholar

Wilkinson, D. J. (2009). Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, **10**(2), 122– 133.
Crossref | CAS | PubMed | Web of Science® | Google Scholar

Wronowska, W., Charzyńska, A., Nienałtowski, K., & Gambin, A. (2015).

Computational modeling of sphingolipid metabolism. *BMC Systems Biology*, **9**(1), 47. https://doi-org.exlibris.colgate.edu/10.1186/s12918-015-0176-9
Crossref | PubMed | Web of Science® | Google Scholar

Number of times cited according to CrossRef: 1

Roberta Coletti, Lorena Leonardelli, Silvia Parolo, Luca Marchetti, A QSP model of prostate cancer immunotherapy to identify effective combination therapies, Scientific Reports, 10.1038/s41598-020-65590-0, **10**, 1, (2020).
Crossref