In the realm of simulation there are many different types of stochastic methods, which one can use to describe a system mathematically, in our case a biological system. These methods do not follow individuals over time, in the case of a biological system “each molecule of a species in the system”, rather they track only on total populations, they assume that the volume of interest is spatially homogenous, which means there are no variations or gradients in the volume being studied. For example if we study a reactor, or in case of system biology a cell, with some species, the assumption will be that the concentration of the molecules is the same throughout the entire volume. There are no localized regions of higher or lower concentration within the reactor or cell.

Mathematical modelling and simulation are essential tools in system biology, there are many reasons and purposes for using the theoretical modelling in system biology, perhaps the most important and reasonable one is testing our assumptions about the system of interest, which then can be used as a reliable gaudiness for scenarios or in case unreliable experimental systems those which are too expensive to experiment.

Terminologies:

* Model: a model is an abstract representation of the system in which we are interest, usually formulated mathematically. This model can, in simple cases, be solved analytically. But in most cases we have to reset to methods to solve the model numerically, which are approximations for analytical methods. The numerical solutions will be more and more accurate as we use smaller and smaller numerical steps.
* Simulation: the numerical solution of a system tries to mimic the behavior of the real system over time, this approach named as simulation. As the British statistician named George Box in 1976 wrote “All models are wrong, some are useful” which means all models are abstractions of reality, simplified versions of the real systems thatthey represent.

Simulation methods:

The choice of simulation method is determined by the aspects of the natural system in which we are interest, our mathematical and computational resources.

* Molecular dynamic simulations: in one hands they are the most accurate models, on the other hand they are the most computationally expensive models. We can categorize these models in two sub groups:

1. Quantum methods: which evaluate the wave functions at the level of individual electrons and are necessary when quantum effects become important.
2. Classical methods: solve the classical equations of motions to simulate the motions for molecules to simulate their motion deterministically.

* Stochastic methods: there are several classes in this category.

1. Spatial discrete stochastic
2. Discrete stochastic

. Master equation

. Stochastic Simulation Algorithm (SSA)

. Tau-leaping

. Chemical Langevin Equation

1. Continuous stochastic

* Continuous deterministic:

1. Random:

. quantum molecular dynamics

. stochastic simulations

1. Deterministic methods:

. classical molecular dynamics

. ordinary differential equations

Our focus here is on the discrete stochastic, which starts with chemical master equation and goes through different numerical methods to approximate the master equation. These methods commonly used in system biology, which can be used to simulate molecular populations over relatively long time periods, which still regarding them as being composed of discrete units.

Using these methods we do not keep track of each molecule or momentum, these methods lose their spatial aspect and must also be assumed that the system of interest is stochastic.

The effects of intrinsic noise have generally been ignored in biology both conceptually and computationally (mathematically). Their state variables are real numbers representing the concentrations of molecules and they do not include noise. These models can be regarded as accurate when we are interested in the mean dynamics of a large number of molecules, large enough that we do not need to worry about individual molecules but can approximate them as populations.

Assume that the system bellow is a biological system with two species which will be converted to each other with the rate constants k1 and k2.

For simulating such system we could in Principe start with the spatial information and velocity of each molecule in the system and then run the simulation and keep track of collisions between molecules and the resulting interactions, however this method is usually too complicated and so computationally too expensive, when the overall number of molecules is large or the dynamics of the system over a long period of time is of interest. Instead of this approach one can ignore spatial information and simply keeps track of the number of molecules of each species, in this case one assumes the system is a homogenous system, in other words, the molecules of each specie are uniformly spread in the system and the system is in thermal equilibrium and the volume of system (cell) is constant.

Because we do not access to the spatial information of the molecules in the system, in each step or time of the simulation, we think in terms of the probability of a reaction taking place, based on the current state of the system, this leads to the “Chemical Master Equation (CME)”, which is a set of ordinary differential equations (ODEs), one ODE equation for each possible state of the system.

represents the probability distribution of the system being in state x at time t. Essentially, it describes the likelihood of finding the system in a particular state xx at a given time t.

The propensity function of reaction j at state xx. It quantifies the likelihood of reaction j occurring at state x per unit time.

The state-change vector associated with reaction j, describing the change in the number of molecules when reaction j occurs. This vector accounts for the stoichiometry of the reaction.

The probability distribution of the system in state x − vj​ at time t. It represents the probability of finding the system in state x − vj after the occurrence of reaction j.

The total number of different types of reactions in the system, capturing the diversity of processes affecting the system's evolution.

Stochastic Simulation Algorithm (SSA):

The direct solution of the Chemical Master Equation (CME) entails calculating the full set of Ordinary Differential Equations (ODEs) to obtain the probability distribution across all possible states for each time (t). However, this approach is computationally demanding, particularly for vast biological systems housing millions or billions of molecules. Instead of directly solving the CME, we can approximate it by sampling from the probability distribution at each time (t). In essence, this involves computing realizations of the state vector {t, x(t)} such that the likelihood of a particular realization being computed reflects the corresponding probability dictated by the CME. This method is known as the Stochastic Simulation Algorithm (SSA) or Gillespie's algorithm. Initially conceived by Joseph L. Doob and others around 1945, it was formally presented by Dan Gillespie in 1976 and gained popularity in 1977 when he efficiently and accurately simulated chemical or biochemical systems of reactions using limited computational resources.

To derive the SSA from the CME, we require the following quantities:

The probability that no reaction takes place in the time interval [t, t + .

The probability that the next reaction occurs in the interval [t + , t and is of type j.

We assume that what occurs over [t, t + τ) is independent of what happens over [t + τ, t + τ + dt), implying that each step in the simulation is independent and unaffected by the previous step. Consequently, we have:

p0()(1-)

Which gives after rearrangement:

With the limit as dτ approaches zero, we obtain:

Now, shifting our focus to , we have:

Here, represents the discrete distribution of the reaction indices j = 1, 2, …, m while describes the density function of an exponential random variable with parameter , characterizing the distribution of waiting time to the next event.

The SSA simulator in “biostoch” library can be used as follow:

Implementation of SSA using biostoch

Tau-leaping Algorithm:

The Stochastic Simulation Algorithm (SSA) ensures exactness in producing statistics from the Chemical Master Equation (CME). However, its practical utility diminishes when reactions occur frequently due to its dynamic update of the entire system after each reaction occurrence. A solution to this inefficiency is to update the system after a certain number of reactions occur within a fixed time interval, denoted as τ. While in some modified versions of the tau-leaping algorithm, τ can be dynamically calculated based on system characteristics, here we consider a fixed interval length.

The tau-leaping algorithm involves updating each reaction in the system based on the number of occurrences within the time interval τ, given x(t). The update equation is:

To ensure accuracy, the time interval τ should be sufficiently small, such that only a relatively small number of reactions take place within it. This restriction ensures that the propensity functions do not change drastically after each update. During the interval [t, t+), remains constant, allowing the number of type-j reactions to fire to be calculated using a Poisson distribution, denoted as In general, a Poisson random variable pp with parameter λ > 0 takes possible values {0, 1, 2, …}.

The tau-leaping algorithm can be implemented using “biostoch” as follows:

Tau-leaping implementation using biostoch

Chemical Langevin Equation (CLE):

if the number of reactions kjkj​ in the tau-leaping approach is sufficiently large for each j, can be approximated by a random distribution with the same mean and variance. To achieve this approximation, each in the tau-leaping algorithm can be replaced with , where Zj​ are independent normal(0, 1) random variables. This transition alters the entries of the state vector XX from natural numbers (representing the number of molecules) to real numbers (denoting molecule amounts in arbitrary units or simply concentration). Consequently, the tau-leaping algorithm can be rewritten as follows, known as the CLE algorithm:

Here, Y(t) represents the state vector in the CLE algorithm, and denotes the noise term.

CLE implementation using biostoch

Reaction Rate Equations (RREs)

If the system is sufficiently large, the right side of the Chemical Langevin Equation (CLE) – the noise term – tends to become negligible compared to the left side. Consequently, one can disregard this term, reducing the equation to a set of ordinary differential equations (ODEs):

This simplification allows for the utilization of various deterministic methods to dynamically simulate the system. Utilizing Euler forward integration, we can derive the RRE model with respect to time, resulting in the following equation:

=

Here, Y(t) represents the deterministic state vector containing concentrations of each species in the system, while vj corresponds to reaction velocities, and aj​ denotes the reaction rate.

Euler implementation using biostoch

The Runge-Kutta equations (RK4)

The Runge-Kutta method, particularly the fourth-order variant (RK4), stands as a notable advancement over the Euler method in numerical integration, especially when dealing with systems characterized by nonlinearity or rapidly changing dynamics. Unlike the Euler method, which approximates the derivative at a single point to update the solution, RK4 calculates intermediate points within each step, leading to greater accuracy and stability.

Mathematically, RK4 employs a weighted average of function evaluations at different points within a time step, resulting in a more accurate estimation of the next state. Specifically, RK4 computes four intermediate slopes using function evaluations at the initial point and three additional points along the trajectory within the time step. These slopes are then weighted and combined to determine the final slope used to update the solution. This multi-step process inherently captures more information about the system's behavior within each time step, leading to improved accuracy compared to the simplistic single-step approach of Euler's method.

Mathematically, we can describe RK4 as follow:

* Calculate the first slope k1​ using the initial condition:
* Use k1​ to estimate the value of Y at the midpoint of the interval:
* Calculate the second slope k2 at the midpoint:
* Use k2​ to make a better estimate of Y at the midpoint:
* Calculate the third slope k3​ at the midpoint:
* Use k3 to make another estimate of Y at the endpoint of the interval:
* Finally, calculate the fourth slope k4​ at the endpoint:
* Combine the slopes to update the solution:

Runge-Kutta implementation using biostoch