STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm

Andrzej M. Kierzek

presentation by Ioan Şucan

Currently:

- huge amount of molecular data is available
- needs computer simulations in order to be studied
- computer software is needed to perform these simulations

Problem setup:

- Fixed volume V
- N chemical species (S_1, S_2, ..., S_N)
- M reactions possible (R_1, ..., R_M)
  - $S_1 + S_2 \longrightarrow S_1S_2$

Question:

Given the number of molecules of each species (X_1, X_2, ..., X_N) at some time t_0, what are the counts of these molecules at any later times ?

Solution:

Use mathematics!

Define the ordinary differential equations (ODEs) of the form

$$\frac{dX_i}{dt} = f_i(X_1, \ldots, X_N)$$

Assumptions:

- $X_i(t)$ is continuous (acceptable for large numbers of molecules)
- reactions evolve as a continuous rate process
- everything is deterministic

Usually the system of ODEs can only be solved numerically.

Issues with presented solution:

- Atom/molecule counts are integers so $X_i(t)$ is not continuous
- The evolution of the system is not deterministic

Instead, we can:

- Assume the system is homogeneous
- Replace the concept of “reaction rate” by “reaction probability per unit time”

This brings us to a *stochastic simulation algorithm*

An exact algorithm for stochastic simulation: *Gillespie's algorithm*

As a first step, define some useful constants:

Let c_u be a reaction and temperature specific constant such that

$c_u\,dt$ = average probability that a particular combination of R_u reactant molecules will interact will react according to R_u in the interval (t, t+dt)

These c_u constants are experimentally determined.

This definition is also regarded as the *fundamental hypothesis* of the stochastic formulation of chemical kinetics.

Evolution of the system:

- Can be done using a “master equation”:

  In essence, the evolution is equivalent to evaluating a large probability density function (pdf):

  $P(X_1, ..., X_N, t)$

  This is rarely solvable.

- Introduce a *reaction probability density function*:

  This should answer the following questions:

  - What is the next reaction that will take place?
  - When will the reaction occur?

The reaction probability density function:
P(T, u) such that,

P(T, u)dt = given some state at time t, the probability that reaction u will take place in the infinitesimal interval (t+T, t+T+dt)

How to compute P(T, u) ?

Remember we have:

$c_u\,dt$ = average probability that a particular combination of R_u reactant molecules will interact will react according to R_u in the interval (t, t+dt)

Introduce h_u = number of distinct reactant combinations for R_u

Define $a_u = h_u\,c_u$

We have defined:

$c_u\,dt$ = average probability that a particular combination of R_u reactant molecules will interact will react according to R_u in the interval $(t, t+dt)$

h_u = number of distinct reactant combinations for R_u

$S_1 + S_2 \longrightarrow \ldots \quad h_u = X_1X_2$

$2S_3 \longrightarrow \ldots \quad h_u = X_3(X_3-1)/2$

$a_u = h_u\,c_u$

This implies:

$a_u\,dt$ = probability that R_u will occur in V during the interval $(t, t + dt)$, given some state at time t

The reaction probability density function can be written as:

$$P(T,u)=P_0(T)a_u dT$$

where

$P_0(T)$ = the probability that no reaction happens in the interval (t, t+T)

The probability that some reaction happens in the interval (t, t + dt):

$$\sum_u a_u dt$$

This implies the probability of no reaction happening is then

$$1-\sum_u a_u dt$$

Notation: $a_0=\sum_u a_u$

$P_0(T+dT)=P_o(T)(1-a_0 dT)$ which implies $P_0(T)=\exp(-a_0 T)$

Previous derivation of P_0 gives P(T, u) to be:

$$P(T,u)=a_u \exp(-a_0 T) \text{ when } 0 \leqslant T < \infty \text{ and } u \in \{1,\ldots,M\}$$
$$P(T,u)=0 \text{ otherwise}$$

Sampling P(T, u) gives a way to decide when the next reaction will occur as well as what that reaction will be.

The only issue that remains is that computers usually have only uniform samples available.

Sampling strategy:

Definition . Let P_X be a distribution on a measure space $(E, \mathcal{B})$. A sequence $X_1, X_2, \ldots$ of random variables is a *sampler for* P_X, if for all $A \in \mathcal{B}$ it holds that

$$P_X(A) = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} 1_A \circ X_i \qquad P\text{-almost surely ,}$$

where $1_A(x) = \begin{cases} 1, \textit{ if } x \in A \\ 0, \textit{ else} \end{cases}$ is the *indicator function* for A.

Sampling strategy: sampling by transformation

If we need to sample according to a pdf g(x) and we have uniform samples Z_i, we can transform Z_i to X_i such that X_i is a sampler for g(x):

First, compute the cumulative density function $\phi: \mathbb{R} \to [0,1]$, which is defined by

$$\phi(y) = \int_{-\infty}^{y} g(u)\, du\,,$$

and its inverse $\phi^{-1}(x)$ [this may be tricky or impossible to do analytically – then, numerical approximations must be called]. Then obtain a sampler X_i from the sampler Z_i by

$$X_i = \phi^{-1}(Z_i)\,.$$

Other methods: rejection, Gibbs, Metropolis

In Gillespie's algorithm:

$P(T,u)=a_u \exp(-a_0 T)$ when $0 \leqslant T < \infty$ and $u \in \{1,\ldots,M\}$

$P(T,u)=0$ otherwise

Assume T, u are independent random variables

$$P(T,u)=P(T)P(u)$$

$$P(T)=a_0 \exp(-a_0 T)$$

$$P(u)=\frac{a_u}{a_0}$$

Use sampling by transformation to get the sampler for P(T,u)

Algorithm pseudocode:

1: Read the constants c_u and the molecule counts X_i

2: Compute a_u and a_0 for the current molecular population

3: Generate r, s uniform random numbers.
Transform them to a sample (T,u) from P(T,u)

$$T = \frac{1}{a_0} \log\left(\frac{1}{r}\right)$$

$$u \in \mathbb{N}, \text{ such that } \sum_{u=1}^{u-1} a_u < s a_0 \leqslant \sum_{u=1}^{u} a_u$$

4: Execute reaction u, increase time by T, adjust the counts X_i

5: If we want to evolve the system further, go to step 2

Additions to the algorithm:

- Allow the volume V to increase linearly
  - Split the evolution into generations.
  - Each generation has duration T; $V(t) = 1 + t / T$
  - Before each step of the algorithm, divide the rates c_u by V(t)

- Simulation of cell division
  - At the end of a generation, reset V to the initial value
  - Divide the number of molecules of each species by 2

- Random pools of reactants
  - If the number of reactants depends on many factors that are hard to model, assume the distribution of their counts is Gaussian and use it for X_i
  - This makes the algorithm *inexact* but it performs well in practice

Implementation (version 1.02):

- mostly C++ code, some Perl code
- command line interface
- output optimized for Gnuplot
- designed to be run as a background process

Simulation of LacZ and LacY genes expression and enzymatic/transport activities of LacZ and LacY proteins

Reaction	Stochastic rate constant [1/s][a]	Meaning
PLac + RNAP → PLacRNAP	0.17	*RNA polymerase binding*/ RNAP—RNA polymerase. PLac—promoter, PLacRNAP closed RNAP/promoter complex
PLacRNAP → PLac + RNAP	10	*RNA polymerase dissociation*
PLacRNAP → TrLacZ1	1	*Closed complex isomerization* TrLacZ1—open RNAP/promoter complex
TrLacZ1 → RbsLacZ + Plac + TrLacZ2	1	*Promoter clearance*. RBSLacZ—RBS, TrLacZ2—RNA polymerase elongating LacZ mRNA
TrLacZ2 → RNAP	0.015	*mRNA chain elongation and RNAP release*
Ribosome + RbsLacZ → RbsRibosome	0.17	*Ribosome binding*. Ribosome—ribosome molecule, RbsRibosome—ribosome/RBS complex
RbsRibosome → Ribosome + RbsLacZ	0.45	*Ribosome dissociation*
RbsRibosome → TrRbsLacZ + RbsLacZ	0.4	*RBS clearance*. TrRbsLacZ—ribosome elongating LacZ protein chain
TrRbsLacZ → LacZ	0.015	*LacZ protein synthesis*
LacZ → dgrLacZ	6.42e−5	*Protein degradation* dgrLacZ—inactive LacZ protein
RbsLacZ → dgrRbsLacZ	0.3	*Functional mRNA degradation*. dgrRbsLacZ—inactive mRNA

Simulation of LacZ and LacY genes expression and enzymatic/transport activities of LacZ and LacY proteins

Quantity	Experimentally determined value[b]	Calculated value[c]
Transcription initiation frequency	0.3 1/s	0.26 1/s
The speed of protein synthesis	20 1/s	22 1/s
Stationary number of mRNA molecules	62	61
Ribosome spacing	110 nucleotides	118 nucleotides

Work done since the paper:

- STOCKS2
  - Better implementation, well organized source code, cross platform
  - Implements “maximal timestep method”
    - Puchalka J. and Kierzek A.M. (2004) Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. Biophys J. 86,1357-1372
    - A combination of:
      - Gibson, M. A. and J. Bruck. 2000. Efficient exact stochastic simulation of chemical systems with many species and many channels. J. Phys. Chem. 104:1876-1889.
      - Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. 115:1716-1733.

For finding the referenced work, please see:

- Gillespie D.T. (1977) Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81, 2340-2361.

- Kierzek A.M. (2002) STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm. Bioinformatics 18, 470-481

- Gibson, M.A. and J. Bruck. 2000. Efficient exact stochastic simulation of chemical systems with many species and many channels. J. Phys. Chem. 104:1876-1889.

- Gillespie, D.T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. 115:1716-1733.

- Puchalka J. and Kierzek A.M. (2004) Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. Biophys J. 86,1357-1372