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 ($R_1, ..., R_M$)
  - $S_1 + S_2 \rightarrow S_1S_2$

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

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

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,

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

How to compute P(T, u) ?

Remember we have:

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

Introduce \( h_u = \text{number of distinct reactant combinations for } R_u \)
Define \( a_u = h_u \, c_u \)
We have defined:

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

h_u = \text{number of distinct reactant combinations for } R_u

S_1 + S_2 \rightarrow \ldots \quad h_u = X_1X_2

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

a_u = h_u \, c_u

This implies:

a_u \, dt = \text{probability that } R_u \text{ will occur in } V \text{ during the interval } (t, t + dt), \text{ 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

\[ P_0(T) = \text{the probability that no reaction happens in the interval (} t, t+T) \]

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: \( a_0 = \sum_u a_u \)

\[ P_0(T + dT) = P_0(T)(1 - a_0 dT) \quad \text{which implies} \quad P_0(T) = \exp(-a_0 T) \]
Previous derivation of $P_0$ gives $P(T, u)$ to be:

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

$$P(T, u) = 0 \quad \text{otherwise}$$

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

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

where $1_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{else} \end{cases}$ 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 $\phi: \mathbb{R} \rightarrow [0,1]$, 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:

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

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 $P(T,u)$
   
   \[
   T = \frac{1}{a_0} \log \left( \frac{1}{r} \right)
   \]
   
   $u \in \mathbb{N}$, such that $\sum_{u=1}^{u-1} a_u < s a_0 \leq \sum_{u=1}^{u} a_u$

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

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Stochastic rate constant [1/s]</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLac + RNAP → PLacRNAP</td>
<td>0.17</td>
<td>RNA polymerase binding/ RNAP—RNA polymerase. PLac—promoter, PLacRNAP closed RNAP/promoter complex</td>
</tr>
<tr>
<td>PLacRNAP → PLac + RNAP</td>
<td>10</td>
<td>RNA polymerase dissociation</td>
</tr>
<tr>
<td>PLacRNAP → TrLacZ1</td>
<td>1</td>
<td>Closed complex isomerization TrLacZ1—open RNAP/promoter complex</td>
</tr>
<tr>
<td>TrLacZ1 → RbsLacZ + Plac + TrLacZ2</td>
<td>1</td>
<td>Promoter clearance. RBSLacZ—RBS, TrLacZ2—RNA polymerase elongating LacZ mRNA</td>
</tr>
<tr>
<td>TrLacZ2 → RNAP</td>
<td>0.015</td>
<td>mRNA chain elongation and RNAP release</td>
</tr>
<tr>
<td>Ribosome + RbsLacZ → RbsRibosome</td>
<td>0.17</td>
<td>Ribosome binding. Ribosome—ribosome molecule, RbsRibosome—ribosome/RBS complex</td>
</tr>
<tr>
<td>RbsRibosome → Ribosome + RbsLacZ</td>
<td>0.45</td>
<td>Ribosome dissociation</td>
</tr>
<tr>
<td>RbsRibosome → TrRbsLacZ + RbsLacZ</td>
<td>0.4</td>
<td>RBS clearance. TrRbsLacZ—ribosome elongating LacZ protein chain</td>
</tr>
<tr>
<td>TrRbsLacZ → LacZ</td>
<td>0.015</td>
<td>LacZ protein synthesis</td>
</tr>
<tr>
<td>LacZ → dgrLacZ</td>
<td>6.42e−5</td>
<td>Protein degradation dgrLacZ—inactive LacZ protein</td>
</tr>
<tr>
<td>RbsLacZ → dgrRbsLacZ</td>
<td>0.3</td>
<td>Functional mRNA degradation. dgrRbsLacZ—inactive mRNA</td>
</tr>
</tbody>
</table>
Simulation of LacZ and LacY genes expression and enzymatic/transport activities of LacZ and LacY proteins

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Experimentally determined value</th>
<th>Calculated value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcription initiation frequency</td>
<td>0.3 l/s</td>
<td>0.26 l/s</td>
</tr>
<tr>
<td>The speed of protein synthesis</td>
<td>20 l/s</td>
<td>22 l/s</td>
</tr>
<tr>
<td>Stationary number of mRNA molecules</td>
<td>62</td>
<td>61</td>
</tr>
<tr>
<td>Ribosome spacing</td>
<td>110 nucleotides</td>
<td>118 nucleotides</td>
</tr>
</tbody>
</table>
Work done since the paper:

• STOCKS2
  • Better implementation, well organized source code, cross platform
  • Implements “maximal timestep method”
• A combination of:
For finding the referenced work, please see:


