GillespieSSA: A user-friendly stochastic simulation package for R

Mario Pineda-Krch
Center for Animal Disease Modeling and Surveillance (CADMS), University of California, Davis
mpineda@ucdavis.edu

Introduction

GillespieSSA is a versatile and extensible framework for stochastic simulation in R and provides a simple interface to a number implementations of the Stochastic Simulation Algorithm (SSA). The methods currently implemented are: the Direct method (D), Explicit tau-leaping (ETL), Binomial tau-leaping (BTL), and Optimized tau-leaping (OTL). The package also provides a library of templates for ecological, epidemiological, and evolutionary continuous-time models that can be customized and extended, e.g. single-species Lotka-Volterra model, Lotka predator-prey, Rosenzweig-MacArthur predator-prey, and Kermack-McKendrick SIR model.

The stochastic simulation algorithm

The stochastic simulation algorithm (SSA) is a procedure for constructing simulated trajectories of finite populations in continuous time. If $X_i(t)$ is the number of individuals in population i ($i = 1, \dots, N$) at time t the SSA estimates the state vector $\mathbf{X}(t) \equiv (X_1(t), \dots, X_N(t))$, given that the system initially (at time t_0) was in state $\mathbf{X}(t_0) = \mathbf{x_0}$. Reactions, single instantaneous events changing at least one of the populations (e.g. birth, death, movement, collision, predation, infection, etc), cause the state of the system to change over time. The SSA procedure samples the time τ to the next reaction R_j ($j = 1, \dots, M$) and updates the system state $\mathbf{X}(t)$ accordingly. Each reaction R_j is characterized mathematically by two quantities;

- its state-change vector $\boldsymbol{\nu}_j \equiv (\nu_{1j}, \dots, \nu_{Nj})$, where ν_{ij} is the change in the number of individuals in population i caused by one reaction of type j and
- its propensity function $a_j(\mathbf{x})$, where $a_j(\mathbf{x})dt$ is the probability that a particular reaction j will occur in the next infinitesimal time interval $[t, t + dt]$.

SSA implementations

There are numerous exact Monte Carlo procedures implementing the SSA. Perhaps the simplest is the Direct method of Gillespie. The Direct method is an exact continuous-time numerical realization of the corresponding stochastic time-evolution equation. Because the Direct method simulates one reaction at a time it is often computationally too slow for practical applications.

Approximate implementations of the SSA sacrifices exactness for large improvements in computational efficiency. The most common technique used is tau-leaping where reaction-bundles are attempted in coarse-grained time increments τ. Speed-ups of several orders of magnitude compared to the Direct method are common. Tau-leaping must be used with care, however, as it is not as foolproof as the Direct method.

Biological examples

Two classical continuous-time models from epidemiology and ecology are presented below as simple examples illustrating how one can implement the GillespieSSA.

Example 1: SIR model

The Kermack-McKendrick SIR model consists of three coupled nonlinear ordinary differential equations,

$$\begin{aligned} dS/dt &= -\beta SI \\ dI/dt &= \beta SI - \gamma I \\ dR/dt &= \gamma I \end{aligned}$$

where at time t, S is the number of susceptible individuals, I the number of infectious individuals, R the number of recovered individuals, β is the infection rate, and γ is the recovery rate. This system consists of the following two reactions,

$$\begin{aligned} R_1 : \; S + I &\xrightarrow{\beta} 2I \\ R_2 : \; I &\xrightarrow{\gamma} R \end{aligned}$$

where the corresponding propensity functions are $a_1 = \beta SI$, and $a_2 = \gamma I$ and state-change matrix (rows are states i and columns are reactions j)

$$\boldsymbol{\nu} = \begin{bmatrix} -1 & 0 \\ +1 & -1 \\ 0 & +1 \end{bmatrix}$$

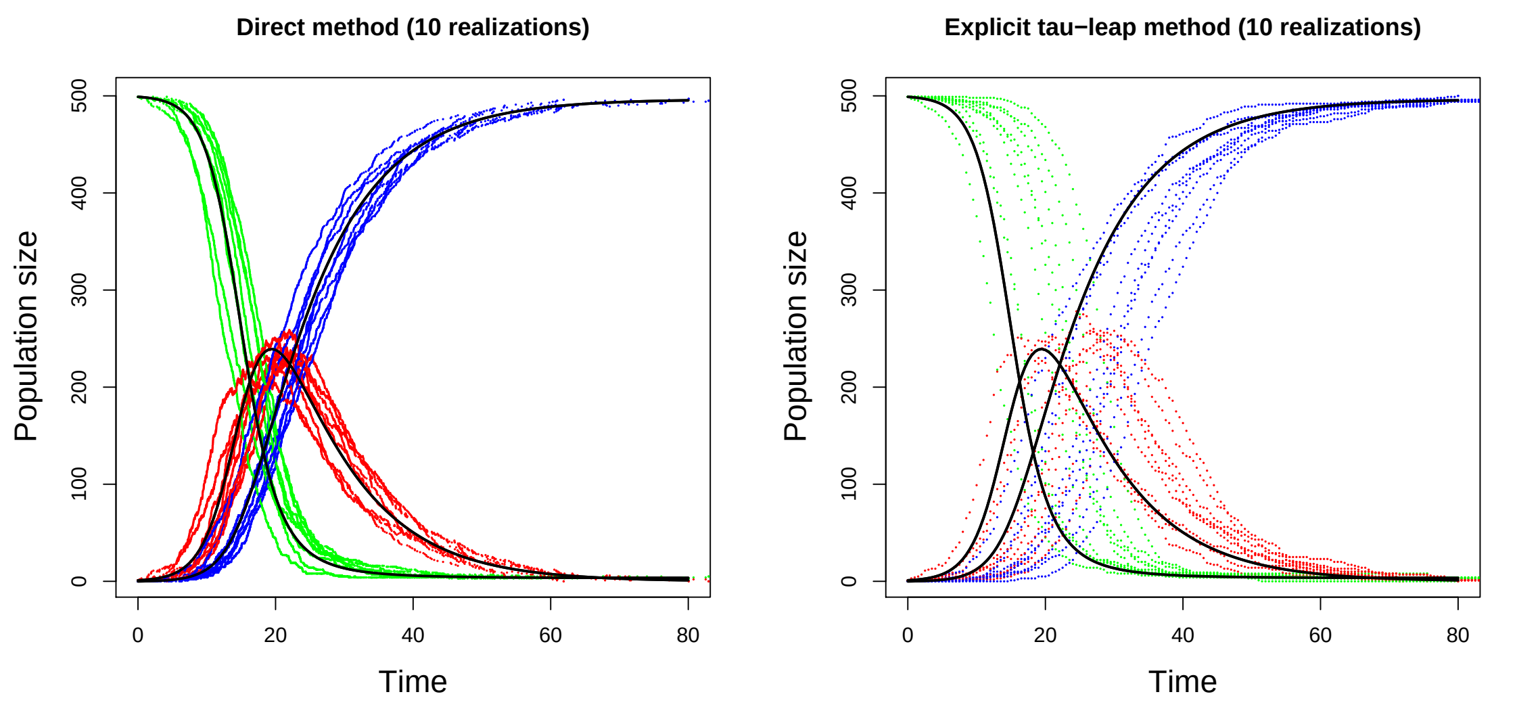
Assuming $\beta = .001$, $\gamma = .1$, $S(0) = 500$, $I(0) = 1$, and $R(0) = 0$ we define the initial state vector, propensity vector, and state-change matrix as

```
R> x0 <- c(S=499,I=1,R=0)
R> a  <- c("0.001*{S}*{I}","0.1*{I}")
R> nu <- matrix(c(-1,  0,
R>                +1, -1,
R>                 0, +1),nrow=3, byrow=T)
```

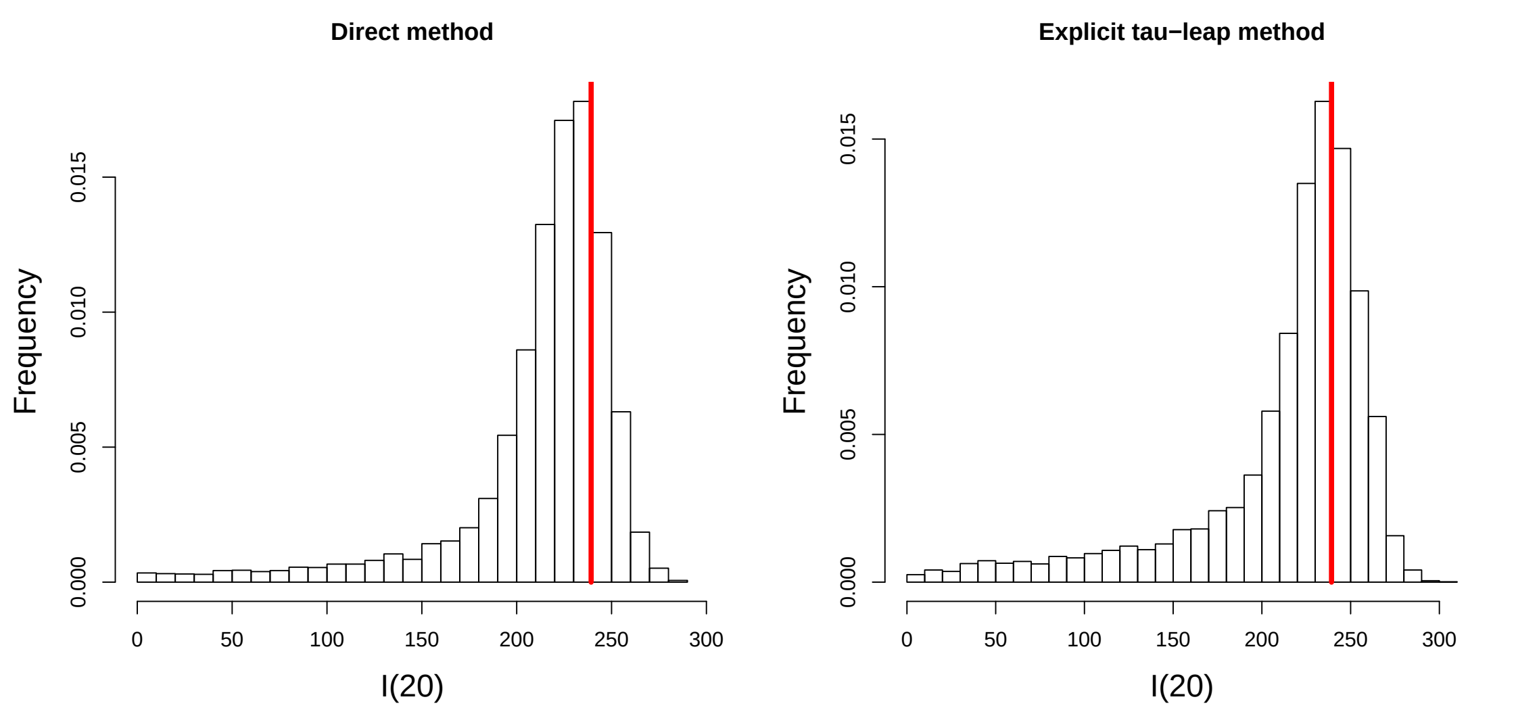
To run a single realization of this system for 100 time units using the default Direct method the following command is issued

```
R> out <- ssa(x0,a,nu,tf=100)
```

The figure below shows the time series for 10 realizations using the Direct and Explict tau-leap methods. Individual points represent single time steps in the time-evolution of the SIR system. The solid curves are the deterministic trajectory of the system. Due to the fixed-sized time leap in the Explicit tau-leap method the points appear more sparse and regularly spaced than in the Direct method.



Running $10,000$ realizations show that the results from the Explicit tau-leap method are virtually indistinguishable from the results of the Direct method and are a good fit to the deterministic prediction (vertical red line in figure).



As a consequence of the larger time-leaps in the Explicit tau-leap method the number of time steps required is dramatically reduced with corresponding reduction in simulation time. The table below illustrates the total duration for the $10,000$ realizations and the median number of time steps per realization. For this particular model, the Explicit tau-leap method is approximately 7 times faster than the Direct metod.

Method	Duration	Median nr of time steps
D	29.7 minutes	993
ETL	4.2 minutes	163

Example 2: predator-prey model

One version of Lotka's predator-prey model is given by

$$\begin{aligned} dY_1/dt &= c_1 Y_1 - c_2 Y_1 Y_2 \\ dY_2/dt &= c_2 Y_1 Y_2 - c_3 Y_2 \end{aligned}$$

consisting of the following three coupled reactions,

$$\begin{aligned} R_1 : \; Y_1 &\xrightarrow{c_1} 2Y_1 \\ R_2 : \; Y_1 + Y_2 &\xrightarrow{c_2} 2Y_2 \\ R_3 : \; Y_1 &\xrightarrow{c_3} \emptyset \end{aligned}$$

where the corresponding propensity functions are $a_1 = c_1 Y_1$, $a_2 = c_2 Y_1 Y_2$, and $a_3 = c_3 Y_2$ and state-change matrix

$$\boldsymbol{\nu} = \begin{bmatrix} +1 & -1 & 0 \\ 0 & +1 & -1 \end{bmatrix}$$

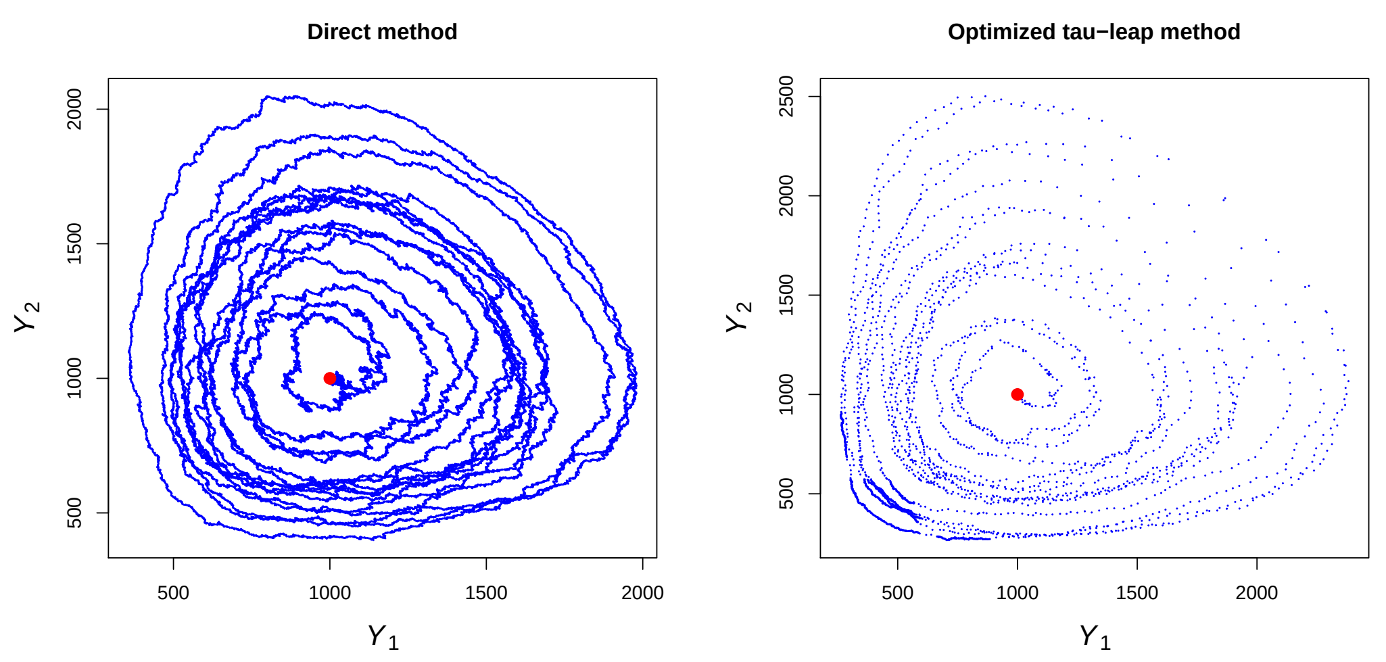
Assuming $c_1 = c_3 = 10$, $c_2 = .01$ and setting $Y_1(0) = Y_2(0) = 1000$ we can now define

```
R> x0<-c(Y1=1000,Y2=1000)
R> a<-c("10*{Y1}",".01*{Y1}*{Y2}","10*{Y2}")
R> nu<-matrix(c(+1, -1,  0,
R>                0, +1, -1), nrow=2, byrow=T)
```

To run the simulation for 100 time units using the Optimized tau-leap method (OTL) we issue the following command

```
R> out <- ssa(x0,a,nu,tf=100,method="OTL")
```

The figure below shows the phase plane trajectory of a single realization of the Direct and the Optimized tau-leap methods where each point represents 5 time steps and the red point indicates the deterministic steady-state (which also is $\mathbf{x}_0$). The Optimized tau-leap method uses an adaptive SSA for it's step size selection, hence the sparse and irregularly spaced points.



In these simulations the Optimized tau-leap method was 54 times faster than the Direct method.

Method	Duration	Nr of time steps
D	10.7 minutes	301353
OTL	12 seconds	14662

Methods

All simulations were performed on a Lenovo laptop with a 1.99GHz Intel Core 2 processor, 2.00GB RAM, running Windows XP and R 2.5.0

Web link

GillespieSSA version 1.0 is avaliable at CRAN and at www.pineda-krch.com/GillespieSSA.

Acknowledgements

Pelayo Alvarez, Tim Carpenter, Mark Carpenter, Heinrich zu Dohna, Rui Lopes, Josh Obrien, Fernando Mardones, Kara O'Keefe, Karen Olmstead, Brant Schumaker, Clair Thunes, Cristobal Verdugo, and Francisco Zagmutt.