Multi-Scale Stochastic Simulation of Diffusion-Coupled Agents and Its Application to Cell Culture Simulation

Yishai Shimoni[1,2]*[9], **German Nudelman**[1]*[9], **Fernand Hayot**[1], **Stuart C. Sealfon**[1]

[1] Department of Neurology and Center for Translational Systems Biology, Mount Sinai School of Medicine, New York, New York, United States of America, [2] Center for Computational Biology and Bioinformatics (C2B2), Columbia University, New York, New York, United States of America

Abstract

Many biological systems consist of multiple cells that interact by secretion and binding of diffusing molecules, thus coordinating responses across cells. Techniques for simulating systems coupling extracellular and intracellular processes are very limited. Here we present an efficient method to stochastically simulate diffusion processes, which at the same time allows synchronization between internal and external cellular conditions through a modification of Gillespie's chemical reaction algorithm. Individual cells are simulated as independent agents, and each cell accurately reacts to changes in its local environment affected by diffusing molecules. Such a simulation provides time-scale separation between the intra-cellular and extra-cellular processes. We use our methodology to study how human monocyte-derived dendritic cells alert neighboring cells about viral infection using diffusing interferon molecules. A subpopulation of the infected cells reacts early to the infection and secretes interferon into the extra-cellular medium, which helps activate other cells. Findings predicted by our simulation and confirmed by experimental results suggest that the early activation is largely independent of the fraction of infected cells and is thus both sensitive and robust. The concordance with the experimental results supports the value of our method for overcoming the challenges of accurately simulating multiscale biological signaling systems.

Citation: Shimoni Y, Nudelman G, Hayot F, Sealfon SC (2011) Multi-Scale Stochastic Simulation of Diffusion-Coupled Agents and Its Application to Cell Culture Simulation. PLoS ONE 6(12): e29298. doi:10.1371/journal.pone.0029298

Editor: Giuseppe Chirico, University of Milano-Bicocca, Italy

Received June 7, 2011; **Accepted** November 23, 2011; **Published** December 21, 2011

Funding: This work was supported by contract HHSN272201000054C and grant U19 AI06231 from the National Institute of Allergy and Infectious Diseases. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: ys2559@c2b2.columbia.edu (YS); german.nudelman@mssm.edu (GN)

[9] These authors contributed equally to this work.

Introduction

Gene expression and signaling events in single cells are stochastic processes. Population measurements and simulations that reflect average cellular responses obscure many aspects of cellular dynamics [1–7]. When stochastic processes are considered, analyses of single cell systems are normally performed assuming that the cells do not interact, and simulations are done using either Gillespie's algorithm [8], or Langevin equations [9]. Gillespie simulations follow the number of molecules present in a cell for several molecular species, and quickly become inefficient as the number of molecules becomes large. Langevin equations can be solved efficiently numerically, but can only be used accurately under a set of restrictive assumptions. Several modifications to the classical Gillespie algorithm have been proposed, some of which lead to more efficient computation [10–13]; others offer parallelization of the algorithm [14,15], and others separation of time scales through the use of hybrid deterministic-stochastic approaches [16,17], or by allowing to estimate the effect of processes that may occur multiple times during a time-step (also known as tau-leaping) [18].

All the techniques mentioned above assume that the cell's state can only change due to processes explicitly defined within the simulation framework. Therefore, when multiple interacting cells are simulated, all cells must be incorporated into a single large simulation, in which identical processes in different cells are counted as different processes. When cells are close to each other, or if the interaction between them is direct in some other way, an efficient compartmental model can be used by including the volume immediately surrounding each cell in the simulation [19]. Such simulations, however, must account for each and every molecule that passes from one compartment to the next, and become unfeasible for large concentrations of signaling molecules.

The computational problem is compounded when the signaling molecules diffuse in the medium and are not directly transferred from cell to cell, since the volume between the cells must be divided into more and more compartments. One way to deal with this problem is to use agent-based models (ABMs), where many individual cells are simulated using the same set of rules (where each agent is a simulation of a single cell). The simulation allows each cell to interact with its local environment. In this case the diffusion process can be simulated separately from intracellular dynamics, allowing separation of time-scales between the internal and external processes.

Such an ABM model was recently developed, in which independent stochastic agents were simulated in conjunction with diffusion processes [20]. However, this work used a deterministic solution to the diffusion equation for cell to cell signaling. Thus, the simulation and resulting analysis ignored the stochastic nature of the interaction between cells. At high concentrations stochasticity may not cause significant variations, but when small concentrations of molecules are involved, an explicitly stochastic approach must be used. Such a simulation, however, that follows the movement of every single molecule is computationally impractical.

Multiple techniques were recently developed to efficiently simulate stochastic reaction-diffusion prcoesses [21–23]. These techniques utilize the statistical properties of independent diffusion processes to allow multiple events of diffusion to occur in each time step (tau-leaping). Notably, one of the methods [24] employs a similar solution to the one we propose here, and may also be applicable to the system we analyze. The solution presented in [24], however, is more complicated conceptually, and more elaborate to implement.

Here we present a method than can be used to stochastically simulate a population of cells that interact by exchanging mediators in a diffusive manner. We first introduce an algorithm that is used to efficiently and stochastically simulate diffusion, and is based on the Monte-Carlo approach. We show that the algorithm leads to random walk behavior for low concentrations, and to dynamics identical to those obtained from the diffusion equation for high concentrations. A second algorithm is introduced, which is a modification of the Gillespie algorithm, and is used to stochastically simulate the internal dynamics of individual cells in a way that ensures a behavior suitable for a Markovian process while allowing changes in external conditions. Diffusing molecules that bind to cell surface receptors activate signaling pathways inside the corresponding cells. The challenge is to match the dynamical intracellular updating as given by Gillespie's algorithm (which is of the order of several minutes), to the time scale associated with diffusion and binding in the extra-cellular medium (which is of the order of several seconds). The modification introduced in the algorithm allows synchronization between the diffusion simulation and the simulation of individual cells at preset time intervals, and is a natural consequence of the Markovian assumption. For convenience we refer to this algorithm as the synchronized Gillespie algorithm. Applying the stochastic diffusion algorithm and the synchronized Gillespie algorithm jointly in an ABM provides an accurate stochastic simulation method for a culture of cells which interact through the diffusion of molecules. As required, this method allows a separation of the time scales between the stochastic diffusion simulation and the synchronized Gillespie algorithm, since the time scales in each one is independent from the other, thus creating a multi-scale simulation.

Specifically, the proposed method allowed us to simulate a culture of monocyte-derived human dendritic cells (DCs), which are the primary response cells mediating the progression from innate to adaptive immunity [25,26]. When a virus such as the Influenza A virus or Newcastle Disease virus infects a DC, the DC is able to detect the infection using the Rig-I protein [27]. This causes the cell to secrete various chemokines and cytokines, of which the most important is interferon. When interferon binds to cell-surface receptors on DCs, it activates the transcription of a large number of genes, including the gene that encodes Rig-I, thereby completing a positive feedback loop that enhances the DCs response to viral infection [28]. A schematics of the mechanism is shown in Fig. 1. Using our simulation method we have recently shown that only a small subset of the cells recognize the viral infection at early times (early responders), and that these cells signal the other DCs and alert them by secreting amounts of interferon that are undetectable experimentally [29]. This small subset is hard to distinguish from experimental noise, and thus simulations become the appropriate method of investigation of this system. Since only a small subset of a population of identical cells cause the response, accurate simulation requires stochastic simulation of individual cells. Additionally, in order to capture the effects of local interferon concentrations at early times, diffusion processes must also be handled stochastically, with allowance made for the high concentrations measured experimentally at longer times following infection. Thus to accurately simulate this system a simulation procedure such as the one proposed is required.

Although DCs are known to adjust the magnitude of response to viral load, the relation of the sensitivity and the control mechanisms to multiplicity of infection (MOI), which is the ratio between the number of viruses and the number of cells in the culture, remains unclear. Here we use our algorithm to investigate how the spreading of cellular response to viral infection depends on MOI. We show that all the infected cells can be activated with similar timing with any MOI of 0.5 or higher, leading to similar interferon production levels. This allows a highly sensitive response to viral infection, which is robust, controlled, and reliable. The results presented here exemplify the value of algorithms for simulations of large systems in which stochastic agents interact via a diffusive process.

Methods

Stochastic Spatially Explicit Diffusion Algorithm

Stochastic simulation of diffusion processes are challenging since they must account for the positions of large numbers of molecules, requiring both large amounts of memory, and strong computational power. One solution to this problem is to consider the local

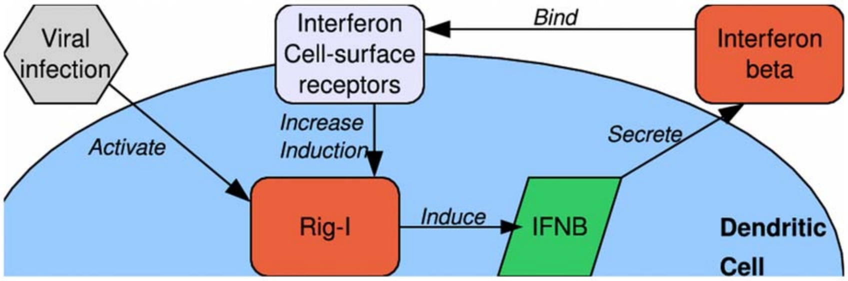

Figure 1. Schematics explaining the positive feedback loop between Rig-I and interferon beta in infected cells. Viral infection is detected by basal levels of Rig-I proteins, which become activated and induce transcription and subsequent secretion of interferon beta. Interferon beta molecules that bind to cell-surface receptors enhance the transcription of Rig-I proteins, thus either completing or activating the positive feedback loop.
doi:10.1371/journal.pone.0029298.g001

concentration of molecules instead of the location of individual molecules, and follow their dynamics using the diffusion equation. Here, the simulation follows the local concentration by dividing the space into voxels (or grid squares in 2D), and following the number of diffusing molecules N per grid square. Using an optimized Monte Carlo algorithm, this allows the underlying random walk behavior of individual molecules while allowing efficient simulation with memory costs that do not depend on the number of molecules in the simulation.

For simplicity we introduce the algorithm in 2D, and later discuss its extension to 3D. We limit the discussion to the experimental scenario described in section I. It should be noted, however, that the simulation can be performed for any set of stochastic entities that interact using secretion and binding of diffusing molecules.

Consider a system in which the agents in the simulation are cells in a medium which we represent as a square lattice. The cells are randomly distributed in the lattice, and periodic boundary conditions are used. We choose the lattice square size to be the size of a single DC, with characteristic length $x_0 = 30\mu m$. Thus, each grid square can either contain a single cell or no cell, and conversely, each cell occupies a single lattice square and does not move throughout the simulation. Additionally, each grid square contains a number of interferon molecules, that can diffuse to neighboring lattice squares with a diffusion coefficient $D = 10^{-11}\frac{m^2}{sec}$ that was determined experimentally [30]. The diffusion algorithm itself is iterative, with each iteration consisting of two steps for each lattice square:

a. For each molecule, decide randomly with a probability p whether within the next time-step it diffuses to a neighboring lattice square.

b. For each diffusing molecule, choose with equal probability the direction in which it diffuses, and move it to the appropriate neighboring square.

As presented above, the algorithm is not efficient since a random number has to be generated for each molecule, which is essentially equivalent to following the movement of every single molecule. To avoid this computationally taxing procedure we reformulate the steps for each grid square as follows:

a. Choose the number of diffusing molecules according to a binomial distribution $N_d = B(N,p)$.

b. Out of the diffusing molecules choose the number of molecules that diffuse north and south according to a binomial distribution $N_{ns} = B(N_d,0.5)$. The rest of the diffusing molecules will diffuse east and west, giving $N_{ew} = N_d - N_{ns}$.

c. Out of N_{ns} choose the number of molecules that diffuse north according to a binomial distribution $N_n = B(N_{ns},0.5)$, thus the number of molecules diffusing south is $N_s = N_{ns} - N_n$.

d. Out of N_{ew} choose the number of molecules that diffuse east according to a binomial distribution $N_e = B(N_{ew},0.5)$, and so the number of molecules diffusing west is $N_w = N_{ew} - N_e$.

This strategy eliminates the need to follow each molecule by choosing a random number for it, and sets the number of random numbers at a constant 4 per lattice square. However, choosing a number from the binomial distribution $B(N,p)$ requires computation time that still depends linearly on the number of molecules N. A simple way to make this process much more efficient is to build a lookup table for the cumulative binomial distribution $C(N,n,p)$. This lookup table holds $C(N,n,p) = \sum_{i=0}^{n}\binom{N}{i}p^i(1-p)^{N-i}$ for every $0 \le n \le N$ and for every $N \le N_{max}$, where N_{max} is the maximal number of molecules found in any lattice square. Thus, a random number can be chosen from a uniform distribution between 0 and 1, and the appropriate location in the sorted lookup table can be efficiently found using a binary search. Using such a lookup table, the execution time of the algorithm scales like $\log_2 N_{max}$, and scales linearly with the area of the simulated system. More accurately, the run-time of each step depends on the sum $\sum_i \log_2 N_i$, where N_i is the local concentration in grid square i. It is reasonable to choose $p = 0.5$, so that a single lookup table can be used both for choosing the number of diffusing molecules and for choosing the number of molecules diffusing in each direction.

It should be noted that relatively efficient approximation methods are available to compute the cumulative probability distribution function for the binomial distribution for any choice of p [31]. The approximation presented in Ref. [31], however, scales like $\sqrt{N}$ and is therefore less efficient than using a lookup table. For large numbers the binomial distribution $B(n,p)$ can be approximated by a normal distribution with an average $\mu = np$ and variance $\sigma^2 = np(1-p)$, which can be calculated efficiently using many existing libraries.

The algorithm to simulate diffusion in 3D can be easily generalized from the 2D procedure presented above. Here, the number of diffusing molecules in each lattice cube will be chosen with some probability p, similar to the 2D case. After choosing the number of diffusing molecules one of six directions has to be chosen. Five random numbers must be chosen for this (e.g. two to choose between axes X, Y, and Z, and one to choose the sign in each axis). A straightforward implementation of this algorithm requires two lookup tables, one for the binomial distribution $B(n,0.5)$ and the other for $B(n,1/3)$. This slightly increases the memory requirements of the algorithm, but retains the property that the computation time scales like $\log_2 N_{max}$, and linearly with the volume of the simulated system. Alternatively, one can use the lookup table with $p = 0.5$ to choose between axis X, Y, Z, and R, (where R stands for 'repeat'), and repeat the process for R iteratively until the number of molecules in direction R is zero. This is a potentially infinite procedure, but on average would require $\frac{1}{2}log_2N$ iterations, and so the total average algorithm run-times will scale like $(log_2N)^2$, while only a single lookup table is required.

Selecting Appropriate Time Step and Lattice Square Size

When simulating diffusion processes, the diffusion coefficient D determines a connection between the time interval τ and the grid size. Normally, one of the parameters (i.e. grid size or time interval) is determined using some external considerations. In our example the lattice grid size was chosen according to the average cell length x_0. To determine the other parameter, the dimensionality of the diffusion coefficient is used to obtain $x_0^2 = \alpha D\tau$, where α is an unknown multiplicative constant that depends on p. For the simulation to be accurate it is necessary to obtain the correct value for α.

This was done by fitting the diffusion simulation algorithm described above to an analytical solution of the diffusion equation, as shown in Fig. 2. Consider the analytically solvable case where at time $t = 0$ all the molecules are located at the origin (using the Kronecker delta function). We initialized the simulation with all the molecules ($N = 2^{20}$) concentrated at the middle of a 201 by 201 lattice, and ran the simulation for 1000 time steps. Due to the high

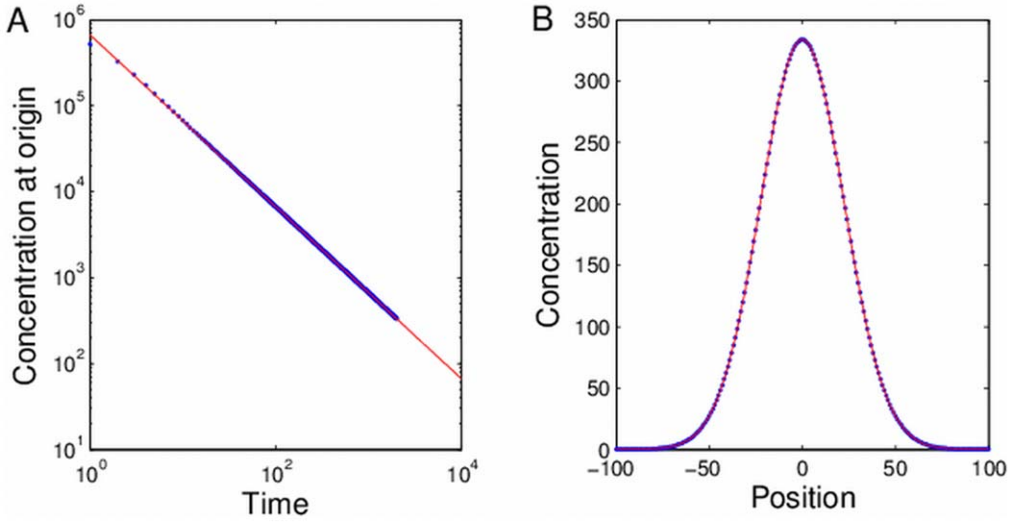

Figure 2. Fitting simulation results to analytical solution. Results of a deterministic diffusion simulation in which at each time step half the molecules at each grid square are equally distributed to neighboring squares. $N=2^{20}$ molecules were placed in the middle of the grid at time $t=0$. A. The concentration in the middle of the grid as a function of simulation steps (dots) and a fit to the inverse of time (solid line). B. The concentration in a horizontal section through the middle of the grid after 2000 simulation steps (dots), and a fit to a Gaussian function (solid line).
doi:10.1371/journal.pone.0029298.g002

number of molecules stochastic effects are negligible, and therefore at each time step we moved exactly one half of the molecules in each grid square, and distributed them equally in all four directions. For the simulation to be considered accurate, the concentration profile of the simulation must follow the solution of the diffusion equation for this condition, which is given by

$$\rho(x,y,t)=\frac{M}{4\pi Dt}e^{-\frac{x^2+y^2}{4Dt}}, \tag{1}$$

where M is the total amount of molecules in the system.

Fig. 2A shows the concentration at the origin as a function of simulation time in a log-log plot (dots), and a fitted A/t function (solid line), where A is the fitted parameter. Except for several steps at the beginning of the simulation, the fitted functions shows a remarkable fit (r-square $R^2>1-10^{-8}$). Fig. 2B shows the concentration of a section through the origin at the end of the simulation (dots), and a fitted Gaussian (solid line), and shows an exceptional fit (r-square $R^2>1-2\cdot10^{-10}$).

The fit presented in Fig. 2A not only shows that the simulation produces the correct dynamics in the functional sense, but also suggests a means to find the appropriate time-length τ to be used in the simulation (given the size of the lattice square x_0 and p). Comparing the fitted function $f(j)$ to the required solution we get

$$f(j)=\frac{A}{j}=\frac{M}{4\pi Dt}. \tag{2}$$

where j is the number of simulation steps. Since $t=j\tau$, we immediately get that $\tau=\frac{M}{4\pi DA}$. M and D are known and A is given by the fit, and so τ can be extracted.

The fit presented above provides an accurate relation between simulation time and real time using the specific values for x_0 and D presented in our system. This can be generalized by remembering that the simulation is independent of D and x_0 and only depends of p, so we can derive the multiplicative constant $\alpha(p)=\frac{x_0^2}{D\tau}$. By substituting the values for x_0, D, and τ we get $\alpha\approx\frac{1}{8}$, or $\tau\approx\frac{x_0^2}{8D}$. We note that this result is true only for a 2D simulation with $p=0.5$. However, the correct coefficient can be easily found in a similar fashion for any value of p, as well as for a 3D simulation. Specifically, for 3D and $p=0.5$ we get $\tau=(\frac{M}{A})^{\frac{2}{3}}\frac{1}{4\pi D}$, and by performing a simulation and fitting the resulting function we get $\tau\approx\frac{0.0818x_0^2}{D}\approx\frac{4x_0^2}{50D}$.

Determining the Fraction of Diffusing Molecules

In some cases both the time step τ and the lattice size x_0 are determined from other considerations. Under such circumstances it is necessary to determine p, the average percentage of molecules that diffuse out of a lattice square within a single time step. The procedure presented in the previous section does not provide a solution for this case, since Eq. 1 provides a relation between x_0 and τ, but not p. An approximate relation between all three variables can be derived by assuming that at time $t=0$ all the molecules are concentrated in the middle of a grid square, and integrating over the solution to the diffusion equation across the lattice square at time $t=\tau$. This gives the proportion of the mass that remains in a grid square after time τ, which is $1-p$. Mathematically, this means solving the equation

$$\frac{1}{4\pi Dt}\int_{-x_0/2}^{x_0/2}\int_{-x_0/2}^{x_0/2}e^{-\frac{x^2+y^2}{4Dt}}dxdy=1-p \tag{3}$$

for p. Using the symmetry of the problem and substituting the integration variable the equation can be re-written in the form

$$\int_0^{\frac{x_0}{\sqrt{4Dt}}}e^{-\eta^2}d\eta=\frac{\sqrt{\pi(1-p)}}{2}. \tag{4}$$

This integral is known as the error function (erf) and cannot be solved analytically, but can easily be solved numerically to find p (and is available as a built-in function in most mathematical libraries).

We note that there are two approximations involved in this analysis. The first is that in the simulation we assume that the molecules are equally distributed (well-mixed) within each grid square, while the mathematical analysis assumes that they are all concentrated at the middle of the square at the start of every time step. The second approximation is that the simulation only allows molecules to diffuse to a neighboring cell at each time step, while the mathematical analysis allows a small percentage to diffuse farther away at that time. In agreement with this analysis, substituting the values found above for x_0 and for τ in Eq. 4, we find a value for p that is close to 0.5 ($p \approx 0.45$). In order to find the accurate value of p simulations are needed in which the value of τ (or x_0) is fixed. As explained in the previous subsections, the values of p and x_0 determine the time step $\tau^*(p)$ that must be used in the simulation (or p and τ determine the size of the lattice square $x^*(p)$). Using half-interval search a value of p can be efficiently found up to arbitrary precision so that $\tau^*(p) = \tau$ (or $x^*(p) = x_0$).

The Synchronized Gillespie Algorithm

The Gillespie algorithm is a simple algorithm that allows accurate simulation of a Markovian process [8]. The algorithm follows the number of molecules for m molecular species involved in n processes. The rate at which process i ($i = 1..n$) occurs is given by $r_i(N_1,..,N_m)$, where N_j ($j = 1..m$) is the number of molecules of type j. The vector $\vec{N} = (N_1,..,N_m)$ represents the state of the system, and the vector $\vec{r}(\vec{N}) = (r_1,...,r_n)$ is the vector holding the transition rates. The fact that the rates of transition between states depend solely on the state itself and not on the simulation history makes it a Markovian process. The algorithm is an iterative one, where at each simulation step the following procedure is performed:

a. Choose the time Δt until the occurrence of the next process from an exponential distribution with an average $r = \sum_{i=1}^{m} r_i$, and advance time by Δt.
b. Choose randomly which process occurs, where each process has a probability proportional to its rate.

The Gillespie simulation results in a different trajectory in phase space at each execution. Still, the statistics of these trajectories are identical to the solution of the master equations for the same system, which are equations describing the dynamics of the probability to find the system in any given state at any given time.

The Gillespie algorithm, however, assumes that the state of the system $\vec{N}$ and the transition rates $\vec{r}(\vec{N})$ can only be changed by the algorithm. Namely, it assumes that every process that can change $\vec{N}$ or $\vec{r}(\vec{N})$ is described by the Gillespie algorithm itself, and not by some other algorithm running in parallel to it. Although it is theoretically possible to incorporate every process within the same Gillespie simulation, such a simulation would become increasingly more complex and inefficient. When integrating a Gillespie simulation into an ABM that allows interactions between agents, the state of the system will invariably be changed by other agents or the environment.

For example, in our case diffusion (which is part of the ABM) changes the number of bound interferon receptors, which affects the virus detection capabilities of the cell through the rate of Rig-I production. In the context of the Gillespie simulation of each cell, the diffusion simulation changes $\vec{N}$ externally at constant intervals, thus breaking the basic assumption of the Markovian process. To account for this, the synchronized Gillespie simulation stops if it tries to update $\vec{N}$ or $\vec{r}(\vec{N})$ after the time at which $\vec{N}$ is supposed to be changed externally (t_u), and advances its internal time to t_u. It then allows the external update of $\vec{N}$ and the resulting rates, and continues from that point on using the newly updated rates. This translates to a modification in the Gillespie updating procedure at each simulation step:

a. Choose the time until the occurrence Δt of the next process from an exponential distribution with an average $r = \sum_{i=1}^{m} r_i$.
b. If the expected time for the next step is larger than the next external state update time $t + \Delta t \geq t_u$ then advance the time to t_u and allow the external state update. Otherwise advance time by Δt, and choose which process occurs, where each process has a probability proportional to its rate.

This modification is an immediate consequence of the Markovian assumption that is at the heart of the algorithm. To show this, we briefly explain the derivation of the exponential distribution for the time-step in the original Gillespie simulation, and then proceed to explain the validity of the modification. We start by assuming that the rate r of a process depends only on the state of the system. As a result, until any process occurs the probability of that process occurring within the next short time interval $P(\Delta t)$ is constant. Therefore, by dividing a long time interval t into n short intervals the probability of nothing happening within t can be written as

$$P(t) = \lim_{n \to \infty} (1 - P(t/n))^n. \tag{5}$$

Since r is the derivative of $P(\Delta t)$ according to time, for short enough time intervals $P(\Delta t) = r\Delta t$, and so Eq. 5 can be rewritten as

$$P(t) = \lim_{n \to \infty} \left(1 - \frac{rt}{n}\right)^n = e^{-rt}. \tag{6}$$

Similarly, consider the case where at a certain time t_u the state of the system changes so that the rate of the process changes to r'. For $t \leq t_u$ the analysis remains as presented above, but for $t > t_u$ the probability $P(\Delta t)$ changes, and so we get

$$\begin{aligned} P(t) &= \lim_{n_1, n_2 \to \infty} \left(1 - \frac{rt_u}{n_1}\right)^{n_1} \left(1 - \frac{r'(t - t_u)}{n_2}\right)^{n_2} \\ &= e^{-rt_u} e^{-r'(t - t_u)}. \end{aligned} \tag{7}$$

Therefore, an initial time t at which the process may occur should be drawn from an exponential distribution with an average of r, as before. If the time t is smaller than t_u then this time should be used. However, if t is larger than t_u (which happens with probability e^{-rt_u}) then a new time t' should be chosen, independendt of t, from an exponential distribution with an average r', and the time at which the process occurs is then $t_u + t'$. This procedure is reflected precisely in the proposed modification to the Gillespie algorithm. We note that this method can be considered a simplified special case of a method presented in [18].

To ensure that this procedure does not introduce artifacts we performed a synchronized Gillespie simulation where at each synchronization step the algorithm stops but no change is made to the state of the simulated system. The times between successive occurrences of the process in such a simulation should display a distribution that is identical to the original Gillespie simulation.

Fig. 3 shows such histograms derived from two simulations involving two processes with rates $r_1 = 1sec^{-1}$ and $r_2 = 0.5sec^{-1}$. The first simulation used the original Gillespie algorithm (dots and X's), and the second one used the synchronized Gillespie algorithm (open circles and open squares). The rates were not changed throughout the simulations, which were performed to a total of 10^7 process occurrences. Both simulations result in identical exponential distribution with the correct mean time between occurrences, showing that the procedure is accurate.

Simulation parameters

Simulation parameters were the same as those previously reported [29]. Below we give a brief description. The modified Gillespie simulation for internal cell dynamics follows the time dependence of interferon transcript (IFN), Rig-I transcripts (DDX) and Rig-I proteins (RIG), which are involved in six reactions: DDX transcription, IFN transcription, RIG translation, DDX degradation, IFN degradation, and RIG degradation. The rate constants for maximal IFN and DDX transcription are given by $K_{IFN} = (20sec)^{-1}$ and $K_{DDX} = (50sec)^{-1}$, respectively. The translation rate constant for RIG is given by $K_{RIG} = (50sec)^{-1}$. The degradation rate constants for IFN, DDX, and RIG are given by $\delta_{IFN} = (10^4sec)^{-1}$, $\delta_{DDX} = (10^4sec)^{-1}$, and $\delta_{RIG} = (10^4sec)^{-1}$, respectively. The transcription rate of Rig-I transcripts depends on the number of bound interferon receptors, denoted by B, in a Michaelis-Menten form using a Hill coefficient of $H_{DDX} = 1.5$ and a half-induction level given by $1/\beta = 75$ (meaning that when $B = 75$ Rig-I reaches half it maximal induction). According to experiments, Rig-I is constitutively expressed in the cells. The ratio between the maximal induction of Rig-I and the constitutive induction is given by $C = 1/50$. Interferon induction depends on Rig-I concentration in a Michaelis-Menten form with a Hill coefficient $H_{IFN} = 3$, and a half-induction concentration given by $1/\gamma = 7000$. All the degradation processes follow an exponential decay.

For the diffusion simulation we use a two-dimensional 40 by 40 square lattice with a square-side length of $L = 30\mu m$ (which is the average diameter of a cell ,as observed experimentally), and holding 200 randomly distributed cells. The diffusion coefficient was determined experimentally to be $D = 10^{-11} \frac{m^2}{sec}$ [30]. The diffusion time step was determined (as explained above) to be $\tau = 9sec$. At each grid square the simulation follows the number of free interferon molecules I, and in squares containing cells also the number of bound receptors B and free receptors F. The initial number of free receptors is $F = 1000$ per cell. At each time step interferon molecules in lattice squares containing cells may bind to free surface receptors, while bound receptors may unbind. The binding rate constant is given by $K_{ON} = (8 \cdot 10^5 sec)^{-1}$, and the unbinding rate constant is $K_{OFF} = (10^3 sec)^{-1}$. We assume that the number of new interferon molecules that are translated at each time step is the number of IFN transcript multiplied by $M = 0.1$. We further assume that interferon secretion is rapid and that all the newly synthesized molecules are secreted at each time step.

Results

We used the proposed ABM simulation to investigate the response to viral infection of monocyte-derived human dendritic cells (DCs), which are the primary response cells that detect infection and trigger the initial innate immune response [25,26]. As depicted in Fig. 1, during the innate immune response, DCs detect viral infection using several receptors and proteins, one of which is Rig-I. Activated Rig-I molecules set in motion a signaling

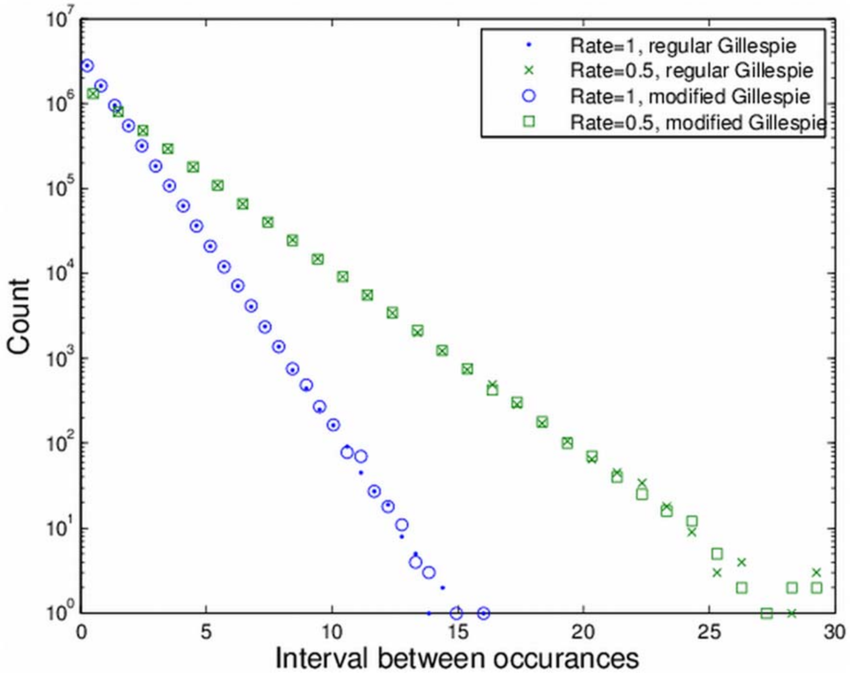

Figure 3. Consistent behavior of the synchronized Gillespie algorithm. Histograms of the intervals between consecutive occurrences of processes in two stochastic simulations. Both simulation were performed with two processes whose rates do not change during the simulation. The first simulation used the original Gillespie algorithm (dots and X's), and the second one used the modified Gillespie algorithm (open circles and open squares) without changing the state of the system during synchronization. Both processes display the same exponential distribution with the correct mean.

doi:10.1371/journal.pone.0029298.g003

cascade that leads to the induction of the interferon beta gene, which upon translation into protein is secreted from the cell. Interferon molecules bind to cell-surface receptors on both infected and uninfected DCs, activating a host of interferon-induced genes, among which is the gene coding for Rig-I. These interactions constitute a positive feedback loop between interferon beta and Rig-I. We recently showed that a fraction of the DCs (less than 1%, the early responder cells) which have sufficient levels of constitutive Rig-I initiate interferon transcription. These cells secrete large amounts of interferon that create locally high concentrations and activate neighboring cells, which in turn produce elevated levels of Rig-I, and if infected subsequently produce and secrete high amount of interferon as well. [29] The ABM proposed here presents a suitable approach to follow the dynamics of the system in a reliable manner, and allows an understanding of the effects of stochasticity on inter-cellular interferon signaling. Fig. 4 shows the levels of Rig-I mRNA vs. interferon mRNA in individual cells, as obtained either from experimental results [29] (left) and from simulation (right) at 6 hrs after infection (top) and 10 hours after infection (bottom).

Although the early response mechanism enables a sensitive yet controlled response to viral infection, the effect of MOI on the early activation of the small sub-population of cells remains unclear. In order to explore the important aspect of robustness of the immune response with changing levels of the multiplicity of infection (MOI), we expanded the previous analysis and performed simulations with varying levels of MOI. In these simulations we examined the efficiency of the immune response activation by the early responder cells. Fig. 5A and Fig. 5B show the levels of interferon and Rig-I mRNA (respectively) as predicted by simulations with MOI = 0.1, 0.5, 1, and 5, which can be compared to the experimental results shown in Fig. 5C and Fig. 5D (the experimental results redrawn from reference [29]). As expected, in a simulation with MOI = 0.1 (dotted lines) the number of early responders that are activated in the simulation is insufficient to efficiently activate the whole population of cells. Most of the cells in the simulation are self activated, and as a result there is hardly any difference between the time of half-maximal induction of interferon and that of the Rig-I gene, (marked by +'s in Fig. 5A and Fig. 5B). On the other hand, it is notable that the lines for MOI = 0.5, 1, and 5 (dash-dotted, dashed, and solid lines, respectively) are very similar. This result suggests that the coordinated response of the cells when activated by early responder cells is highly robust, in the sense that it is both sensitive to low levels of infection, and produces a similarly controlled response even at high levels of infection. The results of these simulation are consistent with the experimental results presented in Fig. 5C and Fig. 5D, where it is shown that increasing the MOI does not result in an increased induction of Rig-I. High noise levels in the experimental results make it difficult to estimate the goodness of the fit at early times (especially for MOI = 0.1, at 5 and 6 hours), but the final result (at 10 hours) is consistent with the simulation predictions, and suggests that the ratio of early responder cells is similar to the one used in the simulation, namely approximately one percent.

Discussion

In this paper we introduced a methodology to combine two stochastic algorithms into an agent based model (ABM) of agents that interact via a diffusive process. The first algorithm allows accurate and efficient stochastic simulation of diffusion processes at both low and high concentrations, and the second allows stochastic simulation of a Markovian system in which the conditions are changed due to external circumstances (i.e. outside the scope of the simulation).

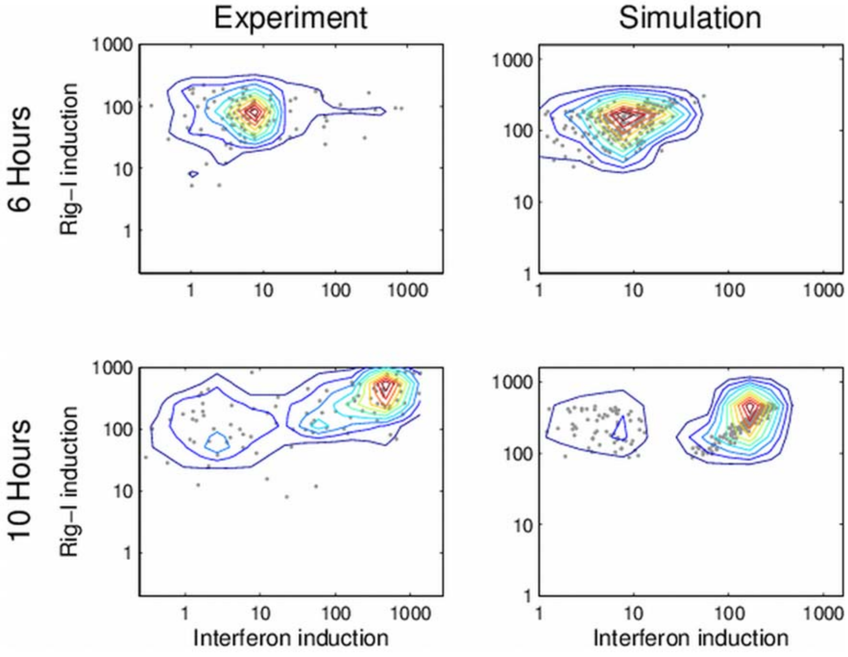

Figure 4. Comparison between experimental results and simulation results of the transcription induction of Rig-I and of interferon. The panels show the amount in individual cells (dots), and an interpolated contour plot of the 2D histogram (solid lines). The two left panels show the experimental results and the two right panels show simulation results. The two top panels and two bottom panels show results obtained at 6 hours and 10 hours, respectively. Significant similarity can be seen between the experimental results and the simulation results. A small population of early responders can be seen in the simulation results at 6 hours, corroborating the early responders hypothesis.
doi:10.1371/journal.pone.0029298.g004

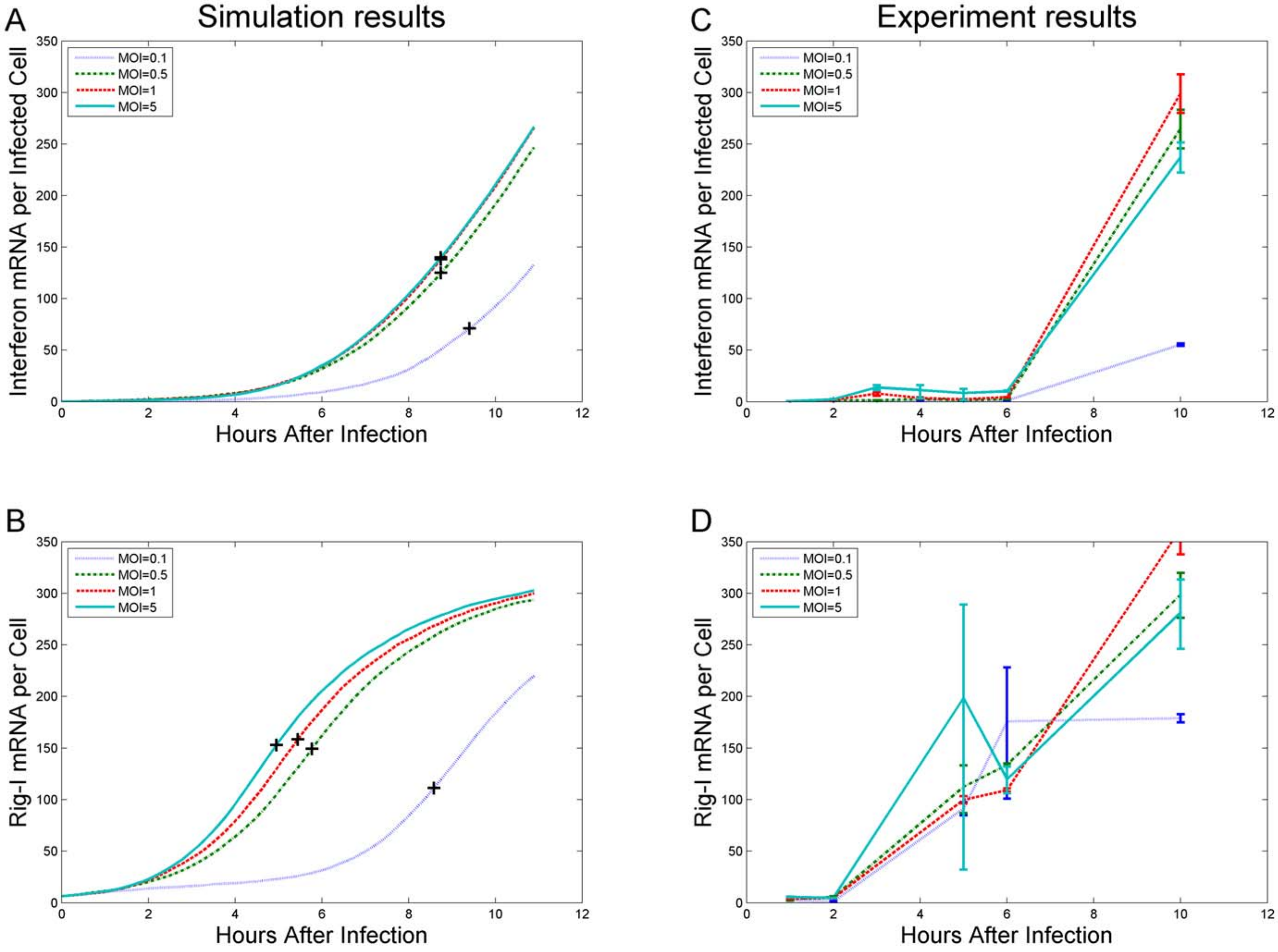

Figure 5. Simulated time-courses of average concentrations for several MOIs and corresponding experimental results. The lines represent the average concentrations of interferon transcripts (panels A and C), and Rig-I transcripts (panels B and D) as obtained by simulation (panels A and B) or from experiments (panels C and D). Both the simulations and the experiments were performed with MOI = 0.1 (dotted lines), MOI = 0.5 (dash-dotted lines), MOI = 1 (dashed lines), and MOI = 5 (solid lines). Specifically, panel A displays the number of interferon mRNA molecules per infected cell as a function of time as predicted by simulation, and panel B presents the number of Rig-I mRNA molecules per cell as predicted by simulation. The time and value of half of the maximal induction is marked by plus markers. Panels C and D show experimental results corresponding to the simulation predictions in panels A and B, respectively. MOI = 0.5, MOI = 1, and MOI = 5 exhibit very similar dynamics, suggesting the activation mechanism is both sensitive and controlled.
doi:10.1371/journal.pone.0029298.g005

The stochastic diffusion algorithm was introduced and was shown to reproduce the required random walk behavior at low concentrations, and to fit the exact solution to the diffusion equation at high concentrations. The algorithm was introduced in 2D and extended to 3D. The fact that the algorithm depends on the probability p of a molecule to diffuse to a neighboring lattice point at every simulation step allows to extend the algorithm to the diffusion of multiple molecular species. To do that, each species receives its own value of p. The value of p for each molecular species can be obtained in a similar way to the one presented above.

The synchronized Gillespie algorithm introduces an important modification to the original Gillespie algorithm, that allows it to accurately run in parallel to other simulations, even when both simulations affect the same molecular species. The proposed modification can be easily applied in combination with various improved versions of the Gillespie algorithm [10–15], allowing for even more efficient and modular simulations.

The usefulness of the combination of these algorithms was demonstrated by applying them for a culture of human DCs that alert each other to viral infection using secretion and diffusion of interferon molecules. The simulation method was previously used [29] to corroborate the hypothesis that a small sub population of the DCs are activated quickly by the infection and alert the other cells, thus expediting the overall immune response. We expanded on these results, and showed that this mechanism of activation is sensitive to small levels of infection (i.e. low MOI), but is highly controlled even for high levels of infection.

It should be noted that our method is not limited to simulation of DCs secreting interferon, or even only to cell cultures, but that it can also be applied to other fields in which agents interact using diffusion processes (e.g. biofilms). Additionally, the modification to the Gillespie algorithm enables precise stochastic simulations of systems that can be affected by external conditions (e.g. externally triggered or time dependent cell division). Finally, the combination of both algorithm allows a separation of time scales between the

Gillespie simulations and the diffusion simulation. Each individual Gillespie simulation uses its own set of reaction rates, allowing for a simulation with multiple time scales.

In conclusion, the methodology presented here paves the way to more extensive analysis of stochastic processes that occur at the scale of cell culture and tissue, or that involve different time scales.

Acknowledgments

We would like to thank Jianzhong Hu and James G. Wetmur for generating the experimental results we refer to in the manuscript, and for their valuable input regarding the application of the algorithm to the biological system.

Author Contributions

Conceived and designed the experiments: YS GN FH SCS. Performed the experiments: YS GN. Analyzed the data: YS GN. Contributed reagents/materials/analysis tools: YS GN. Wrote the paper: YS GN FH SCS.

References

1. Ferrell JE, Machleder EM (1998) The biochemical basis of an all-or-none cell fate switch in Xenopus oocytes. Science 280: 895–898.
2. Elowitz MB, Levine AJ, Siggia ED, Swain PS (2002) Stochastic gene expression in a single cell. Science 297: 1183–1186.
3. Kaern M, Elston TC, Blake WJ, Collins JJ (2005) Stochasticity in gene expression: From theories to phenotypes. Nat Rev Gen 6: 451464.
4. Friedman N, Vardi S, Ronen M, Alon U, Stavans J (2005) Precise temporal modulation in the response of the SOS DNA repair network in individual bacteria. PLoS Biol 3: e238.
5. Hu J, Sealfon SC, Hayot F, Jayaprakash C, Kumar M, et al. (2007) Chromosome-specific and noisy IFNB1 transcription in individual virus-infected human primary dendritic cells. Nucl Acids Res. gkm557 p.
6. Shimoni Y, Altuvia S, Margalit H, Biham O (2009) Stochastic analysis of the SOS response in Escherichia coli. PLoS ONE 4: e5363.
7. Hu J, Iyer-Biswas S, Sealfon SC, Wetmur J, Jayaprakash C, et al. (2009) Power-laws in interferon-b mRNA distribution in virus-infected dendritic cells. Biophysical Journal 97: 1984–1989.
8. Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry 81: 23402361.
9. Gillespie DT (2000) The chemical Langevin equation. The Journal of Chemical Physics 113: 297–306.
10. Gibson MA, Bruck J (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. The Journal of Physical Chemistry A 104: 1876–1889.
11. Slepoy A, Thompson AP, Plimpton SJ (2008) A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. The Journal of Chemical Physics 128: 205101–8.
12. Yang J, Monine M, Faeder J, Hlavacek W (2008) Kinetic Monte Carlo method for rule-based modeling of biochemical networks. Physical Review E 78.
13. Indurkhya S, Beal J (2010) Reaction factoring and bipartite update graphs accelerate the Gillespie algorithm for large-scale biochemical systems. PLoS ONE 5: e8125.
14. Gibson MA, Mjolsness E (2001) Modeling the activity of single genes. In: Bower JM, Bolouri H, eds. Computational Modeling of Genetic and Biochemical Networks. Cambridge, MA: MIT press. pp 1–48.
15. Ridwan A, Krishnan A, Dhar P (2004) A parallel implementation of Gillespies direct method. In: Bubak M, Albada GDv, Sloot PMA, Dongarra JJ, eds. Computational Science - ICCS 2004, Springer Berlin / Heidelberg, volume 3037 of *Lecture Notes in Computer Science*. pp 284–291.
16. Rao CV, Arkin AP (2003) Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. The Journal of Chemical Physics 118: 4999–5010.
17. Kalantzis G (2009) Hybrid stochastic simulations of intracellular reaction-diffusion systems. Com- putational Biology and Chemistry 33: 205–215.
18. Anderson DF (2007) A modified next reaction method for simulating chemical systems with time dependent propensities and delays. J Chem Phys 127: 214107.
19. Elf J, Doncic A, Ehrenberg M (2003) Mesoscopic reaction-diffusion in intracellular signaling. Proceedings of SPIE 5110: 114.
20. Chiam K, Tan C, Bhargava V, Rajagopal G (2006) Hybrid simulations of stochastic reaction-diffusion processes for modeling intracellular signaling pathways. Physical Review E 74.
21. Hattne J, Fange D, Elf J (2005) Stochastic reaction-diffusion simulation with MesoRD. Bioinfor-matics 21: 2923–2924.
22. Marquez-Lago TT, Burrage K (2007) Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics. The Journal of Chemical Physics 127: 104101.
23. Iyengar KA, Harris LA, Clancy P (2010) Accurate implementation of leaping in space: The spatial partitioned-leaping algorithm. J Chem Phys 132: 094101.
24. Drawert B, Lawson MJ, Petzold L, Khammash M (2010) The diffusive finite state projection algorithm for efficient simulation of the stochastic reaction-diffusion master equation. J Chem Phys 132: 074101.
25. Banchereau J, Briere F, Caux C, Davoust J, Lebecque S, et al. (2000) Immunobiology of dendritic cells. Annual Review of Immunology 18: 767–811.
26. Reis e Sousa C (2006) Dendritic cells in a mature age. Nat Rev Immunol 6: 476–483.
27. Kawai T, Akira S (2006) Innate immune recognition of viral infection. Nat Immunol 7: 131–137.
28. Phipps-Yonas H, Seto J, Sealfon SC, Moran TM, Fernandez-Sesma A (2008) Interferon-beta pre-treatment of conventional and plasmacytoid human dendritic cells enhances their activation by inuenza virus. PLoS Pathogens 4: e1000193.
29. Hu J, Nudelman G, Shimoni Y, Kumar M, Ding Y, et al. (2011) Role of cell-to-cell variability in activating a positive feedback antiviral response in human dendritic cells. PLoS ONE 6: e16614.
30. Coppey M, Berezhkovskii AM, Sealfon SC, Shvartsman SY (2007) Time and length scales of autocrine signals in three dimensions. Biophys J 93: 1917–1922.
31. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C, section 6.4. In: Numerical recipes in C, Cambridge University Press. Second edition edition. pp 226–229.