Commun. Comput. Phys.
doi: 10.4208/cicp.130612.121012a

Vol. **14**, No. 2, pp. 461-476
August 2013

An Accelerated Method for Simulating Population Dynamics

Daniel A. Charlebois[1,2,*] and Mads Kærn[1,2,3]

[1] *Department of Physics, University of Ottawa, 150 Louis Pasteur, Ottawa, Ontario K1N 6N5, Canada.*
[2] *Ottawa Institute of Systems Biology, University of Ottawa, 451 Symth Road, Ottawa, Ontario K1H 8M5, Canada.*
[3] *Department of Cellular and Molecular Medicine, University of Ottawa, 451 Symth Road, Ottawa, Ontario K1H 8M5, Canada.*

Received 13 June 2012; Accepted (in revised version) 12 October 2012

Available online 4 January 2013

Abstract. We present an accelerated method for stochastically simulating the dynamics of heterogeneous cell populations. The algorithm combines a Monte Carlo approach for simulating the biochemical kinetics in single cells with a constant-number Monte Carlo method for simulating the reproductive fitness and the statistical characteristics of growing cell populations. To benchmark accuracy and performance, we compare simulation results with those generated from a previously validated population dynamics algorithm. The comparison demonstrates that the accelerated method accurately simulates population dynamics with significant reductions in runtime under commonly invoked steady-state and symmetric cell division assumptions. Considering the increasing complexity of cell population models, the method is an important addition to the arsenal of existing algorithms for simulating cellular and population dynamics that enables efficient, coarse-grained exploration of parameter space.

PACS: 87.10.Mn, 87.10.Rt, 87.16.Yc, 87.17.Ee

Key words: Accelerated stochastic simulation algorithm, constant-number Monte Carlo, gene expression, population dynamics and fitness.

1 Introduction

Cell populations are heterogeneous entities. Part of this heterogeneity arises from the stochasticity inherently present in the process of gene expression, which can result in

[*]Corresponding author. *Email addresses:* `daniel.charlebois@uottawa.ca` (D. A. Charlebois), `mkaern@uottawa.ca` (M. Kærn)

significant variability even among cells with identical genotypes in identical environments [7,15,16,20,26,29,35]. This variability can in turn have significant impact on the overall reproductive fitness of a cell population [1,2,5,9,41,42].

In some cases it is possible to derive analytical solutions for the statistical characteristics of gene expression for simple models (e.g., [25, 27, 30, 31, 36]). However, for more biologically realistic models, these characteristics are available only through numerical simulations. To permit investigations, we previously developed an algorithm for the stochastic simulation of heterogeneous population dynamics at a single-cell resolution [4]. This Population Dynamics Algorithm (PDA) combines the Gillespie stochastic simulation algorithm (SSA) [10,11] to simulate gene expression in individual cells and a constant-number Monte Carlo (MC) method [17,21,22,28,34] for simulating population dynamics.

To benchmark the performance and accuracy of the method, we compared simulation results from the PDA with steady-state and time-dependent analytical solutions for several scenarios, including steady-state and time-dependent gene expression, and the effects on population heterogeneity of cell growth, division, and DNA replication [4]. Additionally, we used the PDA to model gene expression dynamics within bet-hedging cell populations during their adaption to environmental stress. Later, in [5] the PDA and analytical solutions developed for determining the first-passage time dependent fitness of a cell population exposed to a drug over a single generation were found to be in agreement. We refer the reader to these papers for details on the analytical work. These comparisons demonstrated that the PDA accurately captures how complex biological features influence gene expression and population dynamics. However, simulation runtimes can be extensive when the biochemical reaction kinetics that take place within a large number of individual cells are simulated using conventional MC approaches.

To address this problem, we have developed an accelerated method for simulating population dynamics (AMSPD). We first demonstrate that the AMSPD algorithm is numerically accurate and provides a significant speedup compared to the PDA. We then use the AMSPD to perform a parameter scan of a simple model for the development of non-genetic drug resistance to illustrate that it can be advantageous to use the AMSPD and PDA in combination to find an optimal balance between efficiency and accuracy.

2 Algorithm

In this section we present the AMSPD algorithm. The stochastic simulation algorithm [10, 11] and the constant-number MC method [17,21,22,28,34] are also described for completeness.

2.1 Accelerated method for simulating population dynamics

The first step in the AMSPD algorithm is to generate a single stationary time series (such that the moments of the corresponding distribution are not changing) for each biochemical variable in the system using an appropriate simulation method (e.g. the SSA [10,11]

– see Section 2.2) and store the values of the time series in an array of length N. Each row of the array corresponds to a separate biochemical variable. It is not uncommon in simulation studies to assume that one or more biochemical species are in a steady-state (e.g., [1,19,31,37]). The AMSPD algorithm then employs this time series to simulate the gene expression and fitness dynamics of a population of cells (Fig. 1a). Specifically, at the start of the simulation each cell of the initial population is assigned a positive integer (randomly generated from a uniform distribution on the interval $[1,N]$), which corresponds to its column 'position' in the array. During a given sampling interval, each cell progress through the pre-generated time series values stored in the rows of the array. Each time a cell's internal clock is incremented by a pre-specified value time increment Δt, so is its column position in the array (note that the pre-generated time series values were obtained from sampling the SSA simulation using the same Δt). If a cell happens to

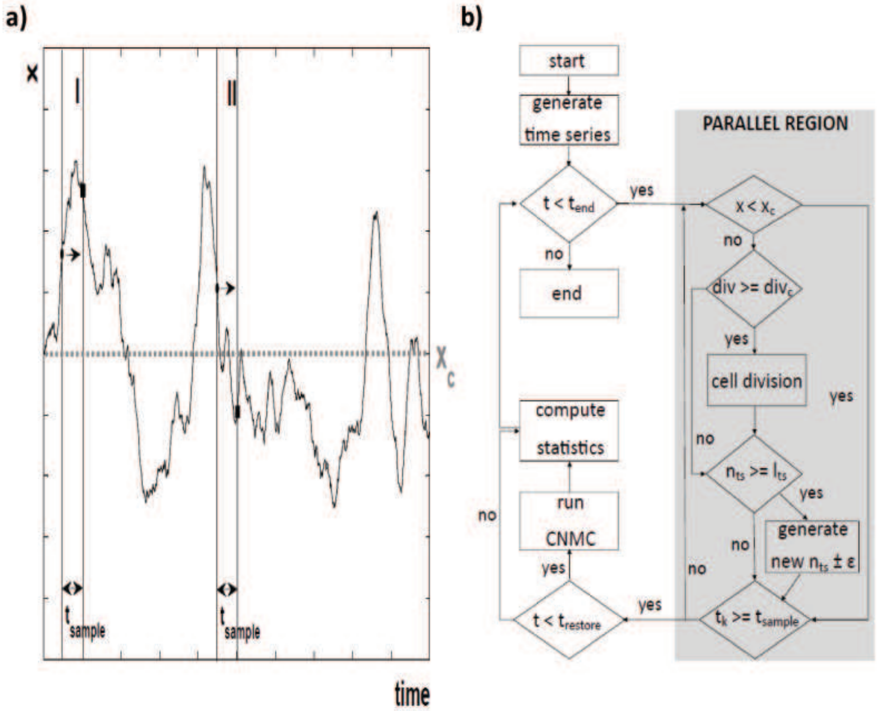

Figure 1: The accelerated method for simulating population dynamics (AMSPD) algorithm. (a) Schematic showing how individual cells are simulated by the AMPSD algorithm. After the cells are randomly assigned positions on the time series (dots), their positions are incremented until the end of the sampling interval t_{sample} is reached (squares). If a reproductive stress is not incorporated into the simulations, then mother cells simply reproduce at a specified rate. However, if the fitness of the cells depends on the level of a particular biochemical variable, then cells can only reproduce if this variable remains above a specified threshold. For instance, in region I, the gene expression value x remains above a critical threshold x_c during the sampling interval and therefore the cell is able to reproduce during the entire interval. In region II, the gene expression value of the cell falls below x_c and is therefore flagged and unable to reproduce after this point. (b) Flow diagram of the AMSPD algorithm presented in the main text (see Table 1 for AMSPD variable and parameter descriptions).

reach the last column before the end of the sampling interval, then the cell is randomly assigned a new position on the array within some error ϵ from the last value, for each biochemical variable in the system. There is a tradeoff between accuracy and efficiency as ϵ is varied (data not shown). For smaller values of ϵ, simulation runtimes are longer but the results are more accurate, and vice versa for larger values of ϵ. In this study we use an ϵ of 10 or lower. For simulations involving the presence of a stressor (e.g. a drug), a biochemical variable of interest (e.g. protein concentration) can be used to determine cellular fitness. For example, if the value of this variable falls below a critical threshold then the cell can be flagged and its biochemical variables no longer simulated nor the cell able to reproduce (Fig. 1a). More elaborate fitness functions than a simple step function can also be incorporated into the AMSPD algorithm. For example, a 'softer' fitness threshold can be modeled using a Hill function with low values of the Hill coefficient n (e.g., $n=2-4$).

Once the end of the sampling interval is reached for all the cells in the population, the constant-number MC method [17,21,22,28,34] is used to keep the number of cells in the population fixed (see Section 2.3). If a cell divides during the sampling interval the concentration of each variable is assumed to remain constant. This is equivalent to assuming that the cellular contents are equally partitioned into equal volumes or that the transient time to steady-state is negligible. This assumption has been used in several other studies (e.g., [3,5,6,18,31]). The daughter cell is then randomly assigned a position on the time series within some tolerance ϵ of each of the mother cell's biochemical variables at the moment of division.

The AMSPD algorithm can be expressed by the flow diagram (Fig. 1b) and the sub-

Table 1: AMSPD parameters and variables.

Parameter/Variable	Description
div and div_c	Division variable and corresponding threshold at which division occurs.
ϵ	Error term for assignment or re-assignment of position on the stationary time series.
l_{ts}	Length of the stationary time series.
$NC_{daughter}$	Number of daughter cells born in a given sampling interval.
$NC_{population}$	Total number of cells in the population.
n_{ts}	Position on the stationary time series.
t	Global simulation time.
t_{end}	Simulation end time.
t_k	Local or cell specific simulation time.
t_{sample}	Sampling interval for statistics.
$t_{restore}$	Interval between population size restores.
x and x_c	Biochemical variable of interest and the corresponding threshold below which cells are unable to reproduce.

sequent pseudocode (Algorithm 2.1). In the pseudocode the AMSPD parameters and variables are defined as follows: div is the division parameter (generally time or volume) and div_c the corresponding threshold (if applicable) at which division occurs, l_{ts} the length (number of points) of the time series, t the global simulation time, t_{end} the user specified simulation end time, t_k the local or cell specific simulation time, t_{sample} the sampling interval for statistics, $t_{restore}$ the interval between population size restores, $NC_{daughter}$ the number of daughter cells, $NC_{population}$ the total number of cells in the population, n_{ts} the position on the time series, x a biochemical variable of interest and x_c the corresponding threshold (if applicable) below which cells are unable to reproduce. The AMSPD parameters and variables for pseudocode the are summarized in Table 1.

Algorithm 2.1: AMSPD

```
Generate a stationary time series for each variable using the SSA (see Algorithm 2.2)
Randomly obtain an initial $n_{ts}$ for each cell
while $t < t_{end}$ do
  begin parallel region
  for all $NC_{population}$ such that $t_k < t_{sample}$ do
    Update $t_k$ and $div$
    if $x \geq x_c$ then
      Update $n_{ts}$ and $x$
      if $n_{ts} \geq l_{ts}$ then
        Randomly generate new $n_{ts}$ (until $x(n_{ts})$ within $\pm\epsilon$ of $x(l_{ts})$) and update $x$
      end if
      if $div \geq div_c$ then
        Execute cell division
        Increment $NC_{daughter}$
      end if
    end if
  end for
  end parallel region
  Update $t$ and $t_{sample}$
  Execute constant-number MC (see Algorithm 2.3)
  Compute statistics
end while
```

2.2 SSA

In the Direct Method Gillespie SSA [10, 11], M chemical reactions with rate constants $c_1,\cdots,c_M$ among N chemical species $X_1,\cdots,X_N$, are simulated one reaction event at a time. The next reaction to occur Ω and its timing Γ are determined by calculating M reaction propensities $a_1,\cdots,a_M$, given the current number of molecules of each of the N chemical species, to obtain an appropriately weighted probability for each reaction. It can be implemented via the following pseudocode:

Algorithm 2.2: SSA

```
if $t < t_{end}$ and $\alpha_0 = \sum_{v=1}^{M} a_v \neq 0$ then
    for $v = 1,M$ do
        Calculate $\alpha_v$
    end for
    $\alpha_0 = \sum_{v=1}^{M} a_v$
    Generate uniformly distributed random numbers $(r_1,r_2)$
    Determine when $(\Gamma = \ln(1/r_1)/\alpha_0)$ and which $(\min\{\Omega \mid \alpha_\Omega \geq r_2\alpha_0\})$ reaction will occur
    Set $t = t + \Gamma$
    Update $X_1,\cdots,X_N$
end if
```

2.3 Constant-number Monte Carlo method

The constant-number MC method [17,21,22,28,34] permits the statistically accurate simulation of a representative sample of an exponentially growing cell population. In this implementation of the method, all the daughter cells born since the last update $NC_{daughter}$ are stored and simulated using a separate array from the mother cells. To avoid simulating the daughters of daughter cells, the interval between population size updates $t_{restore}$ is chosen such that mother cells divide at most once, and daughter cells not at all, during a particular $t_{restore}$ interval. The constant-number MC method can be represented by the following pseudocode:

Algorithm 2.3: CNMC

```
if $t > t_{restore}$ and $NC_{daughter} \geq 1$ then
    for all $NC_{daughter}$ do
        Randomly select mother cell
        Replace mother cell with oldest available daughter cell
    end for
end if
```

3 Numerical results and discussion

To evaluate the accuracy and the speedup of the accelerated method, we compare simulation results obtained using the AMSPD algorithm to those obtained using the previously validated PDA [4].

For benchmarking we first examine a univariate model of protein production and decay (Section 3.1). Then we consider a multivariate model of gene expression where mRNA and protein production and decay are both incorporated (Section 3.2). To further benchmark the algorithm when the reproductive fitness of the cell population in the presence of a drug is incorporated, we reproduce the results from [5]. In this work, we used an Ornstein-Uhlenbeck (OU) model to simulate gene expression (Section 3.3). Finally, in Section 3.4, we demonstrate that the accelerated method can enable efficient and numerically accurate coarse-grained exploration of the parameter space corresponding to a population model. Specifically, we use the AMSPD algorithm to perform a scan of the parameter space of the OU model of gene expression, and compare the resulting fitness landscape of the population with results obtained using the PDA.

Both algorithms were implemented in Fortran 90 and executed on an IBM with 2 quad-core processors (1.86GHz cores) and 2.0GB of RAM. All units unless indicated otherwise are arbitrary. Statistics were estimated from 10 realisations of populations consisting of 1000 cells unless otherwise indicated.

3.1 Univariate model

We consider gene expression as a birth-death process modeled by the following equations

$$\varnothing \xrightarrow{k_P} P, \tag{3.1}$$

$$P \xrightarrow{\delta_P} \varnothing, \tag{3.2}$$

where P is a protein produced in a single step at a rate k_P (Eq. (3.1)), and decays at a rate δ_P (Eq. (3.2)).

We first model cell division without incorporating cellular volume, that is each cell divides once its cellular 'clock', or time since last division div, reaches or exceeds a predefined cell division time div_c. In this case, excellent agreement is found between the AMSPD algorithm and PDA (Fig. 2a). The runtime of the AMSPD algorithm is shown in Fig. 2b. When the time to generate the time series for the AMSPD algorithm is incorporated into the runtime of the AMSPD algorithm, then the AMPSD's runtime increases linearly with k_P. However, if the time to generate the time series is not factored into AMSPD's runtime, then the runtime of the AMSPD algorithm does not vary with k_P since a time series of the same length is used in each of the simulations. This applies for instance if a time series that was previously generated can be used again, for example, if the simulation is to be repeated, or if a variable assumed not to affect gene expression (such as div_c) is changed. The AMSPD algorithm is found to be significantly faster as the rate of protein production is increased (Fig. 2c). For instance, when k_P is 1 the AMSPD algorithm is three times faster than the PDA. However, when k_P is increased to 100, the speedup is sixty times. If the time to generate the time series for the AMSPD algorithm is not factored into the speedup calculation, then speedups of several hundred times are observed. We attribute the speedup to the fact that the AMSPD algorithm does not simulate every reaction occurring inside each cell of the population as the PDA does. Rather, the AMSPD algorithm performs a random access lookup in an array containing the values of the time series.

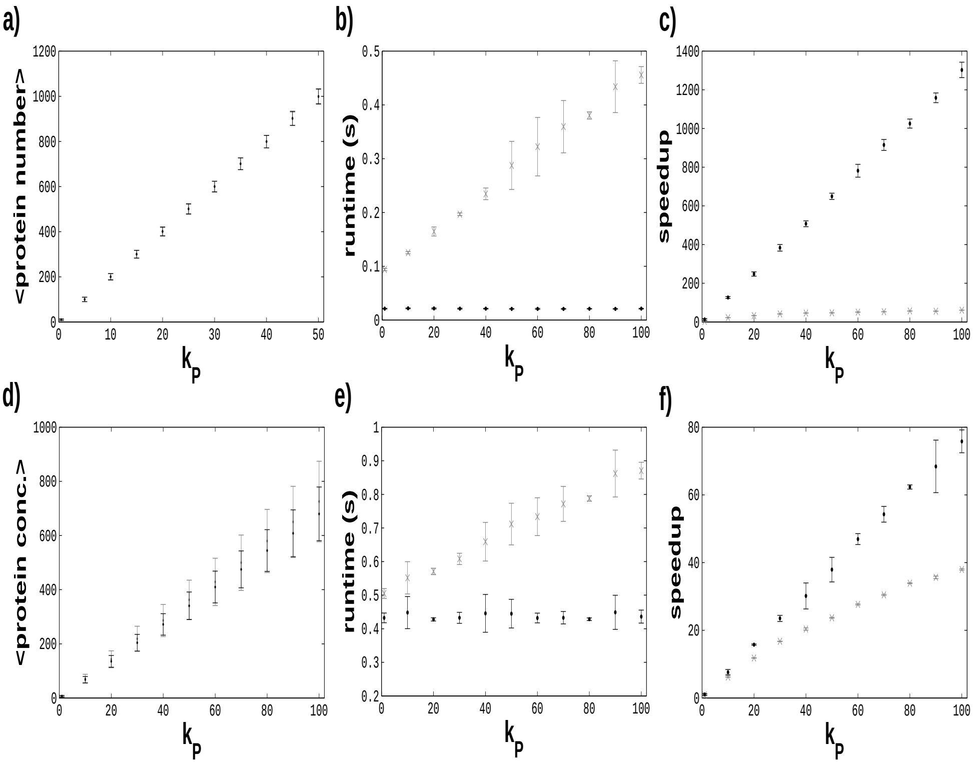

Figure 2: Comparison of accuracy and performance of the AMSPD algorithm and PDA [4] for a birth-death model of gene expression. Panels (a)-(c) correspond to simulation results for volume independent cell division and (d)-(f) volume dependent cell division. (a) and (d) show the average steady-state protein numbers and concentrations, respectively, as a function of the rate of protein production k_P for the AMSPD algorithm (gray) and the PDA (black). (b) and (e) show the runtime of the AMSPD simulation. (c) and (f) show the speedup of the AMSPD algorithm, when compared to the runtime of the PDA, as a function of the rate of protein production k_P. Gray x's in (b) and (e), and in (c) and (f), are the results obtained when the time to produce the gene expression time series is incorporated into the AMSPD's runtime and the speedup calculation, respectively. Black dots in (b) and (e), and in (c) and (f), are the results obtained when the time to produce the gene expression time series is not incorporated into the AMSPD's runtime and the speedup calculation, respectively. Simulations were started from steady-state ($p^s = k_P/\delta_P$), the initial time since last division div drawn from a uniform distribution [0,div], and the protein time series generated by the AMPSD algorithm contained 10^4 values. The parameters were set to $\delta_P = 0.01$, $\epsilon = 10$, $div_c = 100$, and $t_{end} = 1000$.

The incorporation of changing cellular volume throughout the cell cycle into simulations can be important when concentration dependent, rather than absolute number, effects are to be considered (e.g., [36, 40]). In a more complex model, we describe cell growth by an exponential growth law [4, 12, 13]

$$V_k(t_{div}) = V_0 2^{(t_{div}/\tau_0)}, \tag{3.3}$$

where V_0 is the cell volume at the time of its birth, and τ_0 is the interval between volume doubling. Cell division occurs when the cell volume reaches $2V_0$.

Again, there is excellent numerical agreement between the two simulation methods (Fig. 2d), and a significant speedup when using the AMSPD algorithm (Fig. 2f). For example, when k_P is 1 the speedup is thirteen times. However, when k_P is increased to 100, the AMSPD algorithm is about forty times faster than the PDA, and seventy five times faster when the time to generate the time series for the AMSPD algorithm is not included in the speedup calculation. As in the previous case (Fig. 2b), AMPSD's runtime either increases linearly with k_P or does not vary with k_P, depending on whether the time to generate the time series for the AMSPD algorithm is or is not incorporated into the runtime, respectively (Fig. 2e). The runtimes shown in Fig. 2e are longer than those in Fig. 2b due to the incorporation of cellular volume dynamics.

The results presented in this section indicate that the AMSPD algorithm can accurately simulate models that incorporate a univariate description of biochemical dynamics occurring inside of growing and dividing cells with a significant reduction in runtime when compared to the PDA.

3.2 Multivariate model

The previous section considered a univariate analysis. However, it is in the multiple variate scenario that the AMSPD algorithm is likely to be employed since any model incorporating a biologically realistic level of detail will require more than one variable. Due to the nonlinearity and dimensionality of the corresponding system of equations, a computational approach rather than an analytical one will generally be required to obtain solutions. However, as the dimensionality of the system increases so does the computation time along with the need for an accelerated simulation approach.

In order to benchmark the AMSPD algorithm in the multivariate case, we use a slightly more complex model where gene expression is simulated as a two-step process described by the following equations

$$D \xrightarrow{k_M} D+M, \tag{3.4}$$

$$M \xrightarrow{k_P} M+P, \tag{3.5}$$

$$M \xrightarrow{\delta_M} \emptyset, \tag{3.6}$$

$$P \xrightarrow{\delta_P} \emptyset, \tag{3.7}$$

where Eqs. (3.4)-(3.5) respectively describe the transcription and translation processes. The degradation of mRNA M and protein P are accounted for by Eqs. (3.6)-(3.7), respectively.

As in the univariate case, we consider volume independent (Fig. 3a-3d) and volume dependent cell division (Fig. 3e-3h). Excellent agreement is found between the algorithms for mRNA and protein steady-states (Fig. 3a and 3e, and Fig. 3b and 3f, respectively). AMPSD's runtime again either increases linearly with k_P or does not vary with k_P, depending on whether the time to generate the time series for the AMSPD algorithm

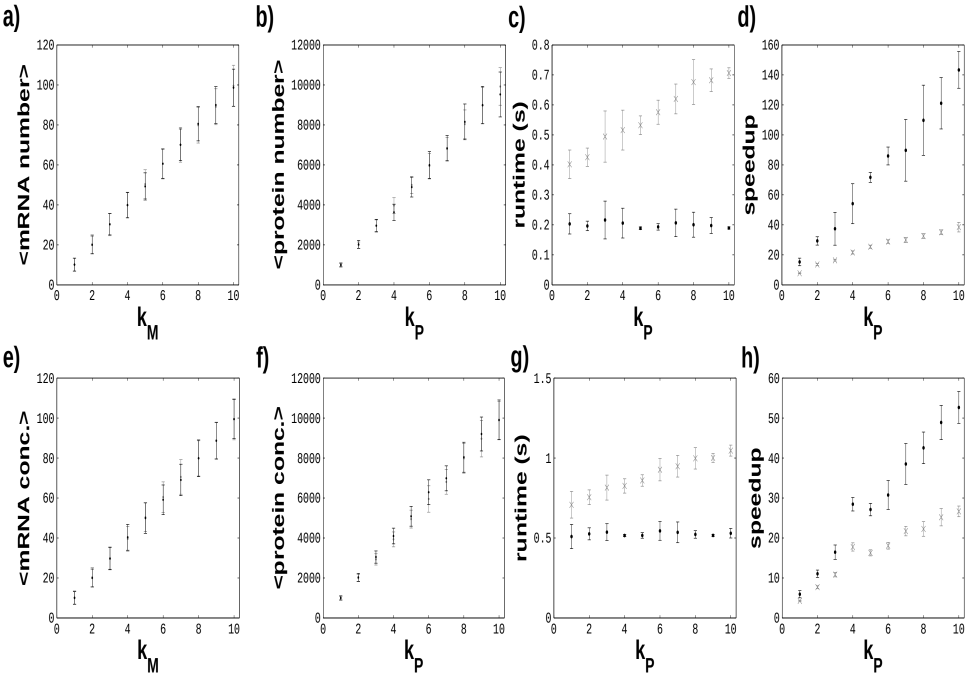

Figure 3: Comparison of accuracy and performance of the AMSPD algorithm and PDA [4] for a two-step model of gene expression. Panels (a)-(d) correspond to simulation results for volume independent cell division and (e)-(h) volume dependent cell division. (a) and (e) show the steady-state mRNA numbers and concentrations, respectively, as a function of the rate of mRNA production k_M for the AMSPD algorithm (gray) and the PDA (black). (b) and (f) show the average steady-state protein numbers and concentrations, respectively, as a function of the rate of protein production k_P for the AMSPD algorithm (gray) and the PDA (black). (c) and (g) show the runtime of the AMSPD simulation. (d) and (h) show the speedup of the AMSPD algorithm, when compared to the runtime of the PDA, as a function of the rate of protein production k_P. Gray x's in (c) and (g), and in (d) and (h), are the results obtained when the time to produce the gene expression time series is incorporated into the AMSPD's runtime and the speedup calculation, respectively. Black dots in (c) and (g), and in (d) and (h), are the results obtained when the time to produce the gene expression time series is not incorporated into the AMSPD's runtime and the speedup calculation, respectively. Simulations were started from steady-state ($M^s = k_M/\delta_M$ and $P^s = k_M k_P/\delta_M \delta_P$), the initial time since last division div drawn from a uniform distribution [0,div], and the protein time series generated by the AMPSD algorithm contained 10^4 values. The parameters were set to $k_M = 1$ (when k_P was varied), $k_P = 1$ (when k_M was varied), $\delta_M = 0.1$, $\delta_P = 0.01$, $\epsilon = 10$, $div_c = 100$, and $t_{end} = 1000$.

is or is not incorporated into the runtime, respectively (Fig. 3c and 3g). The AMSPD algorithm is significantly faster especially when the rate of protein production was high. For instance, considering volume independent division when k_P is 10, the AMSPD algorithm is roughly forty times faster than the PDA, and one hundred and forty times faster when the time to generate the time series for the AMSPD algorithm is not factored into the speedup calculation (Fig. 3d). When volume dependent division is incorporated and k_P is 10, the AMSPD algorithm is twenty five times faster than the PDA, and fifty five times faster when the time to generate the time series for the AMSPD algorithm is not included in the speedup calculation (Fig. 3h).

Together the results in this section demonstrate that the AMSPD algorithm can be extended to accurately and efficiently simulate multivariate biochemical networks when cell growth and division are incorporated into the model.

3.3 Environmental stress

In this section we use the AMSPD algorithm to reproduce the results obtained in [5] using the PDA to simulate the reproductive fitness of a cell population exposed to a drug. In that study, gene expression in individual cells was simulated as an OU process to capture the effect of fluctuations in gene expression x on the development of drug-resistant cell populations. It was found that if the fluctuation relaxation time scale in gene expression (non-genetic memory) was sufficiently long then drug resistant population could emerge independently of genetic mutations (genetic memory) [5]. The range of values for the non-genetic memory parameter for which drug resistance emerged independently of mutations was in agreement with 'mixing time' (defined as the lag where the autocorrelation function has decreased by 50%) results found experimentally in a human lung cancer cells [33].

The OU process can be described by the following Langevin equation

$$\frac{dx(t)}{dt}=\frac{1}{\tau}(\mu-x(t))+c^{1/2}\xi_t, \tag{3.8}$$

where c and τ are the diffusion constant and the relaxation time, respectively, and ξ_t is Gaussian white noise ($\langle\xi_t\rangle=0$, $\langle\xi_t\xi_{t'}\rangle=\delta(t-t')$) [38]. Without loss of generality, we set the mean μ equal to zero and use the fluctuation time-scale τ to model the time-scale of non-genetic memory.

As in [5], 'microfitness' $w(x)$ describes the effect of a drug on the reproductive fitness of individual cells with a given level of expression. For simplicity, in this model microfitness is described using a Heaviside step function, such that a cell is unable to reproduce if their expression level is below a critical value, $w(x<x_c)=0$, and unaffected by the drug otherwise, $w(x\geq x_c)=1$. For the OU process with a mean of zero, 50% of the cell population is instantaneously unable to reproduce when the drug is applied at generation zero. The 'macrofitness' W, or reproductive fitness of the cell population, is here calculated by dividing the number of cells that reproduced during a specified sampling interval by the total number of cells (held fixed by the constant-number MC method) in the population. Since we have set the cell division time such that each cell can only divide once during a given sampling interval, the maximum macrofitness that the cell population can attain is one.

Fig. 4 illustrates that as the number of generations increase, the cell population will reach a steady-state level of fitness. The level of drug resistance that the cell population develops depends on the degree of non-genetic memory. When the non-genetic memory is sufficiently low (i.e. $\tau<=1$) the population completely succumbs to the drug, and

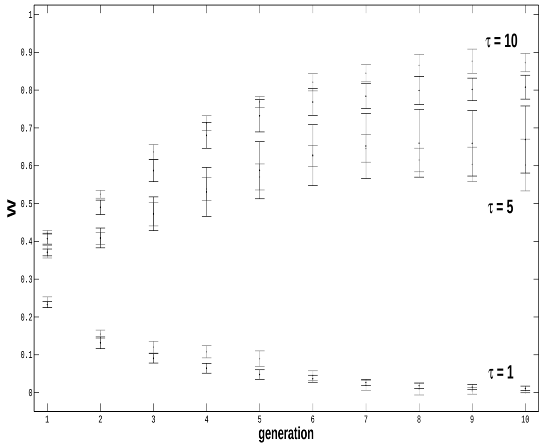

Figure 4: Comparison of accuracy of the AMSPD algorithm (gray) and PDA (black) [4] for a model capturing the effect of non-genetic memory τ on drug resistance at various timescales [5]. The reproductive fitness of the cell population (macrofitness) W as a function of generation is plotted for various values of τ. Simulations were started from the steady-state OU distribution (with mean $\mu=0$ and variance $\sigma^2=c\tau/2=1$), the initial time since last division div was drawn from a uniform distribution [0,div], and the protein time series generated by the AMPSD algorithm contained 10^6 values. The parameters were set to $\epsilon=1$ and $div_c=1$, and scaled by div_c. The threshold below which cells were unable to reproduce x_c was set to μ.

when non-genetic memory is sufficiently high ($\tau>1$) the macrofitness of the cell population increases (Fig. 4). This phenomenon occurs because higher values of τ have a higher probability of enabling individual cells to reside for sufficiently long times in advantageous regions of the fitness landscape, such that they can reproduce before succumbing to the effects of the drug. These results are in quantitative agreement with results previous obtained using the PDA algorithm [5] and demonstrate that the AMSPD algorithm can be used to simulate more biologically complex scenarios such as the effect of stress and noisy gene expression on the reproductive fitness of a cell population.

3.4 Parameter scans

In order to investigate the dynamics of a given population model, one can perform simulations across the corresponding parameter space. However, the use of a more accurate method, such as the PDA, to perform these simulations can prohibit a comprehensive parameter scan due to its computationally intensive nature. The use of an approximate method such as the AMSPD algorithm can enable an efficient preliminary exploration of the parameter space.

Using the OU model of gene expression and the framework presented in Section 3.3 to capture fitness dynamics, we simulate the reproductive fitness of a cell population after being exposed to a stress for 10 generations.

In Section 3.3 the variance of the OU distribution was fixed to 1 by varying the diffusion constant c as the relaxation time τ was increased. Here, τ and c are varied indepen-

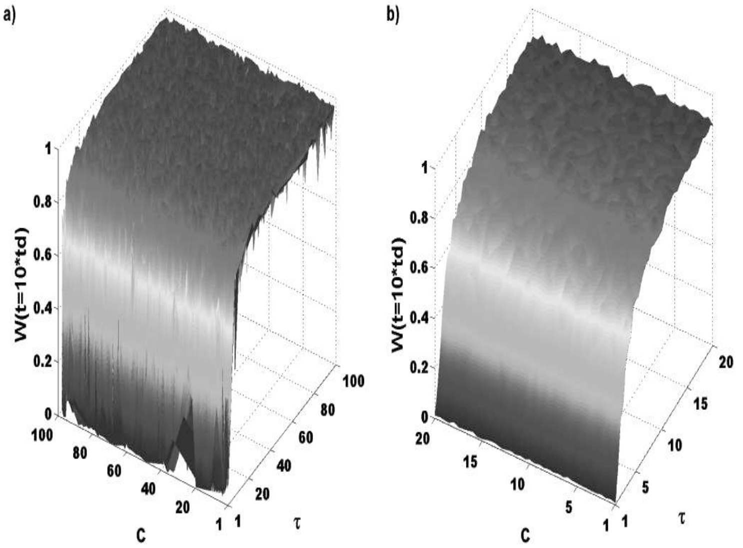

Figure 5: Parameter scans of an OU model of gene expression for the development of drug resistance. (a) Stochastic simulations carried out using the AMSPD algorithm. Here, 10^4 parameter combinations for the relaxation time τ and the diffusion constant c were generated using a Latin hypercube sampling method [23, 24], in order to determine the reproductive fitness of the cell population (macrofitness) W after 10 generations. (b) A systematic scan of a region of the parameter space shown in (a) using the more accurate PDA [4]. Simulations were started from the steady-state OU distribution (with mean $\mu = 0$ and variance $\sigma^2 = c\tau/2$), the initial time since last division div was drawn from a uniform distribution $[0, div]$, and the protein time series generated by the AMPSD algorithm contained 10^6 values. The parameters were set to $\epsilon = 1$ and $div_c = 1$, and scaled by div_c. The threshold below which cells were unable to reproduce x_c was set to μ.

dently to further examine the role that these parameters have on fitness. Using a Latin hypercube sampling method [23, 24], we generate 10^4 different parameter combinations and simulate the population dynamics using AMSPD (Fig. 5a). Based on the results of these simulations we then identify a region of parameter space of interest (reduced by a factor of 5 compared to the original parameter space), namely where the macrofitness of the cell population changes rapidly, and then perform the simulations using the PDA (Fig. 5b). The parameter scans show that in this model the diffusion constant does not affect population fitness independently of τ (Fig. 5a and 5b).

The fitness landscapes obtained using the two methods are qualitatively in agreement (Fig. 5a and 5b). This suggests that the AMSPD algorithm can be used to efficiently identify coarse parameter regimes, which can then be further refined by more accurate simulation using the PDA.

4 Conclusion

We have presented an accelerated method for simulating cellular population dynamics. The method generates and employs single representative time series to simulate the gene expression and reproductive fitness dynamics of all the cells in the population. A constant-number MC method [17,21,22,28,34] is used by the AMPSD algorithm in order to simulate a statistically representative sample of an exponentially growing cell population. This approach allows for accurate simulations with a significant speedup compared to simulations obtained using a previously developed population dynamics algorithm [4]. The accelerated algorithm is a course-grained method designed for scenarios when all the variables of an intracellular biochemical reaction network can be assumed to be at steady-state and cells to divide symmetrically (e.g., [3,8,14,39]). In order to reduce the complexity of the model and simulation times, these assumptions are often invoked when simulating gene expression and cellular dynamics (e.g., [1,3,5,6,18,19,31,37]). Although these assumptions are not always biologically realistic, due to speed of the accelerated method, efficient scans over a large parameter space can be performed in order to identify regions of interest. Once the parameter space region of interest is identified, simulations can then be performed using a more accurate population simulation algorithm. Correspondingly, this method should prove useful for the simulation of gene expression and population models of ever increasing complexity. Furthermore, it is anticipated that the method will apply more generally to other scenarios, for example, to speedup simulations of biochemical reaction networks during periods when the rate parameters are not varying due to noise external to the system [32].

Acknowledgments

This work was supported financially by the National Science and Engineering Research Council of Canada (NSERC).

Author contributions

D.C. conceived and designed the research. D.C. developed the algorithm and performed the simulations. D.C. and M.K. wrote the manuscript. M.K. supervised the study.

References

[1] M. Acar, J. T. Mettetal and A. van Oudenaarden, Stochastic switching as a survival strategy in fluctuating environments, Nat. Genet., 40 (2008), 471-475.

[2] W. Blake, G. Balazsi, M. Kohanski, F. Isaacs, K. Murphy, Y. Kuang, C. Cantor, D. Walt and J. Collins, Phenotypic consequences of promoter-mediated transcriptional noise, Mol. Cell, 24 (2006), 853-865.

[3] B. M. Boman, M. S. Wicha, J. Z. Fields and O. A. Runquist, Symmetric division of cancer stem cells – a key mechanism in tumor growth that should be targeted in future therapeutic approaches, Clinical Pharmacology and Therapeutics, 81 (2007), 893-898.

[4] D. A Charlebois, J. Intosalmi, D. Fraser and M. Kaern, An algorithm for the stochastic simulation of gene expression and heterogeneous population dynamics, Commun. Comput. Phys., 9 (2011), 89-112.

[5] D. A Charlebois, N. Abdennur and M. Kaern, Gene expression noise facilitates adaptation and drug resistance independently of mutation, Phys. Rev. Lett., 107 (2011), doi: 10.1103/PhysRevLett.107.218101.

[6] A. S. Ribeiro, D. A. Charlebois and J. Lyold-Price, *CellLine*, a stochastic cell lineage simulator, Bioinformatics, 23 (2007), 3409-3411.

[7] A. Eldar and M. Elowitz, Functional roles for noise in genetic circuits, Nature, 467 (2010), 167-173.

[8] B. Feierbach and F. Chang, Roles of the fission yeast formin for3p in cell polarity, actin cable formation and symmetric cell division, Curr. Biol., 11 (2001), 1656-1665.

[9] D. Fraser and M. Kaern, A chance at survival: gene expression noise and phenotypic diversification strategies, Molec. Microbiol., 71 (2009), 1333-1340.

[10] D. T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comput. Phys., 22 (1976), 403-434.

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

[12] T. Lu, D. Volfson, L. Tsimring and J. Hasty, Cellular growth and division in the Gillespie algorithm, Syst. Biol., 1 (2004), 121-128.

[13] D. Volfson,, J. Marciniak1, W. J. Blake, N. Ostroff1, L. S. Tsimring and J. Hasty, Origins of extrinsic variability in eukaryotic gene expression, Nature, 439 (2006), 861-864.

[14] W. B. Huttner and Y. Kosodo, Symmetric versus asymmetric cell division during neurogenesis in the developing vertebrate central nervous system, Curr. Opin. Cell. Biol., 17 (2005), 648-657.

[15] M. Kaern, T. C. Elston, W. J. Blake and J. J. Collins, Stochasticity in gene expression, Nat. Rev. Genet., 6 (2005), 451-464.

[16] B. B. Kaufmann and A. van Oudenaarden, Stochastic gene expression: from single molecules to the proteome, Curr. Opin. Genet. Dev., 17 (2007), 107-112.

[17] Y. Lin, K. Lee and T. Matsoukas, Solution of the population balance equation using constant-number Monte Carlo, Chem. Eng. Sci., 57 (2002), 2241-2252.

[18] T. Lu, D. Volfson, L. Tsimring and J. Hasty, Cellular growth and division in the Gillespie algorithm, Syst. Biol., 1 (2004), 121-128.

[19] D. Nevozhay, R. M. Adams, E. V. Itallie, M. R. Bennett and G. Balazsi, Mapping the environmental fitness landscape of a synthetic gene circuit, PLoS Comput. Biol., 8 (2012), doi:10.1371/journal.pcbi.1002480.

[20] N. Maheshri and E. K. O'Shea, Living with noisy genes: how cells function reliably with inherent variability in gene expression, Annu. Rev. Biophys. Biomol. Struct., 36 (2007), 413-434.

[21] N. V. Mantzaris, Stochastic and deterministic simulations of heterogeneous cell population dynamics, J. Theor. Biol., 241 (2006), 690-706.

[22] N. V. Mantzaris, From single-cell genetic architecture to cell population dynamics: Quantitatively decomposing the effects of different population heterogeneity sources for a genetic network with positive feedback architecture, Biophys. J., 92 (2007), 4271-4288.

[23] M. D. McKay, R. J. Beckman and W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics, 21 (1979), 239-245.

[24] M. D. McKay, Sensitivity and uncertainty analysis using a statistical sample of input values, in: Y. Ronen (Ed.), Uncertainty Analysis, Ch. 4, pp. 145-186, CRC Press, Bcca Raton, Florida, 1988.

[25] R. Murugan, Multiple stochastic point processes in gene expression, J. Stat. Phys., 131 (2008), 153-165.

[26] J. Paulsson, Summing up the noise in gene networks, Nature, 427 (2004), 415-418.

[27] J. M. Raser and E. K. O'Shea, Control of stochasticity in eukaryotic gene expression, Science, 304 (2004), 1811-1814.

[28] D. Ramkrishna, The status of population balances, Rev. Chem. Engng., 3 (1985), 49-95.

[29] M. S. Samoilov, G. Price and A. P. Arkin, From fluctuations to phenotypes: The physiology of noise, Sci. STKE, 366 (2006), re17.

[30] M. Scott, B. Ingalls and M. Kaern, Estimations of intrinsic and extrinsic noise in models of nonlinear genetic networks, Chaos, 16 (2006), 026107.

[31] V. Shahrezaei and P. S. Swain, Analytical distributions for stochastic gene expression, PNAS, 105 (2008), 17256-17261.

[32] V. Shahrezaei, J. Ollivier and P. Swain. Colored extrinsic fluctuations and stochastic gene expression, Mol. Syst. Biol., 4 (2008), 196.

[33] A. Sigal, R. Milo, A. Cohen, N. Geva-Zatorsky, Y. Klein, Y. Liron, N. Rosenfeld, T. Danon, N. Perzov and U. Alon, Variability and memory of protein levels in human cells, Nature, 444 (2006), 643-646.

[34] M. Smith and T. Matsoukas, Constant-number Monte Carlo simulation of population balances, Chem. Eng. Sci., 53 (1998), 1777-1786.

[35] J. L. Spudich and D. E. Koshland, Non-genetic individuality: chance in the single cell, Nature, 262 (1976), 467-471.

[36] P. S. Swain, M. B. Elowits and E. D. Siggia, Intrinsic and extrinsic contributions to stochasticity in gene expression, PNAS, 99 (2002), 12795-12800.

[37] M. Thattai and A. van Oudenaarden, Attenuation of noise in ultrasensitive signaling cascades, Biophys. J., 82 (2002), 2943-2950.

[38] G. Uhlenbeck and L. Ornstein, On the theory of Brownian motion, Phys. Rev., 36 (2008), 823-841.

[39] S. Woolner and N. Papalopulu, Spindle position in symmetric cell divisions during epiboly is controlled by opposing and dynamic apicobasal forces, Dev. Cell, 22 (2009), 775-787.

[40] R. Zadrag-Tecza, M. Kwolek-Mirek, G. Bartosz and T. Bilinski, Cell volume as a factor limiting the replicative lifespan of the yeast Saccharomyces cerevisiae, Biogerontology, 10 (2009), 481-488.

[41] Z. Zhang, W. Qian and J. Zhang, Positive selection for elevated gene expression noise in yeast, Mol. Syst. Biol., (2009), doi:10.1038/msb.2009.58.

[42] D. Zhuravel, D. Fraser, S. St-Pierre, L. Tepliakova, W. Pang, J. Hasty and M. Kaern, Phenotypic impact of regulatory noise in cellular stress-response pathways, Syst. Synth. Biol., 4 (2010), doi:10.1007/s11693-010-9055-2.