BMC Systems Biology

BioMed Central

Methodology article

A Dominated Coupling From The Past algorithm for the stochastic simulation of networks of biochemical reactions

Martin Hemberg[1] and Mauricio Barahona*[1,2]

Address: [1]Department of Bioengineering, Imperial College London, South Kensington Campus, London SW7 2AZ, UK and [2]Institute for Mathematical Sciences, Imperial College London, South Kensington Campus, London SW7 2AZ, UK

Email: Martin Hemberg - martin.hemberg03@imperial.ac.uk; Mauricio Barahona* - m.barahona@imperial.ac.uk

* Corresponding author

Published: 8 May 2008

BMC Systems Biology 2008, **2**:42 doi:10.1186/1752-0509-2-42

Received: 20 September 2007

Accepted: 8 May 2008

This article is available from: http://www.biomedcentral.com/1752-0509/2/42

Abstract

Background: In recent years, stochastic descriptions of biochemical reactions based on the Master Equation (ME) have become widespread. These are especially relevant for models involving gene regulation. Gillespie's Stochastic Simulation Algorithm (SSA) is the most widely used method for the numerical evaluation of these models. The SSA produces exact samples from the distribution of the ME for finite times. However, if the stationary distribution is of interest, the SSA provides no information about convergence or how long the algorithm needs to be run to sample from the stationary distribution with given accuracy.

Results: We present a proof and numerical characterization of a Perfect Sampling algorithm for the ME of networks of biochemical reactions prevalent in gene regulation and enzymatic catalysis. Our algorithm combines the SSA with Dominated Coupling From The Past (DCFTP) techniques to provide guaranteed sampling from the stationary distribution. The resulting DCFTP-SSA is applicable to networks of reactions with uni-molecular stoichiometries and sub-linear, (anti-) monotone propensity functions. We showcase its applicability studying steady-state properties of stochastic regulatory networks of relevance in synthetic and systems biology.

Conclusion: The DCFTP-SSA provides an extension to Gillespie's SSA with guaranteed sampling from the stationary solution of the ME for a broad class of stochastic biochemical networks.

Background

Recent experiments on gene and enzyme activity at single cell resolution have revealed the inherent randomness of key cellular processes linked to gene expression [1-3]. The experiments show that populations with identical genotypes present heterogeneous phenotypes and that noise at the molecular level, due to low copy numbers, contributes to population diversity. For mathematical models to capture this variability, a stochastic description is required.

Stochastic models in Computational Biology are usually based on the Master Equation (ME) of the chemical reaction kinetics [4-6]. Formally, the ME is a differential form of the Chapman-Kolmogorov equation, which gives the time evolution of $P(\mathbf{x}, t)$, the probability of the state of the system $\mathbf{x}$. Only a handful of analytical solutions of the ME have been found and one must usually resort to approximations or numerical solutions. The most popular numerical procedure is Gillespie's Stochastic Simulation

Algorithm (SSA) [7,8], a kinetic Monte Carlo algorithm that provides exact stochastic realizations of the underlying system of reactions. Each run of the SSA produces a time trace for the system; a collection of independent runs can be used to obtain convergent statistics of the time-dependent solution of the ME. In many situations, one is interested in the steady state properties of the system, i.e., in the stationary distribution of the ME, π. Although in principle π could be obtained as the first left eigenvector of the transition matrix, this computation is infeasible for most problems of interest due to the combinatorial explosion of the state space [9]. To circumvent this problem, it has become customary to sample π by running the SSA for a 'very long time', convincing oneself through different heuristics that stationarity has been attained. However, the SSA does not provide guarantees or information about how long the algorithm must run to converge to π. In recent years there has been an increased interest in finding algorithms which can address the issue of sampling from stationarity, e.g., a strategy based on forward flux sampling [10].

In a seminal paper in the field of Markov Chain Monte Carlo, Propp and Wilson introduced the idea of *Coupling From The Past* (CFTP), an ingenious procedure that provides *guaranteed* sampling from the stationary distribution of a Markov chain by running coupled chains from all possible initial conditions from the past [11]. Algorithms that guarantee sampling from the stationary distribution of a Markov chain are referred to as *Perfect Sampling* algorithms [11-14]. Recently [15], we introduced a Perfect Sampling algorithm for the SSA of biochemical networks based on Kendall's Dominated CFTP (DCFTP) [13]. This paper expands on our previous work by providing an explicit implementation of the algorithm together with a mathematical proof of its applicability to a class of reactions prevalent in models of gene regulation. We also study its numerical properties through a series of expanded examples drawn from Systems and Synthetic Biology.

Preliminaries and definitions

Dominated Coupling From The Past (DCFTP)

We give here a brief introduction to the CFTP framework (see [11-13] for full proofs).

The central idea behind CFTP is to find a time in the past such that the whole state space is mapped to the same state at the present, for a given sequence of random numbers. When that occurs, the state at the present can be considered to be a sample of the stationary distribution. More formally, consider a Markov process defined by the transition rule $\mathbf{X}_{t+1} = \varphi(\mathbf{X}_t)$, where $\mathbf{X}_t \equiv \mathbf{x}(t)$ is shorthand for the state of the system at time t. Any Markov chain $\mathbf{X}_{-T}^{-\infty} \equiv \{\mathbf{X}_{-\infty}, \ldots, \mathbf{X}_{-T}\}$, started from $t = -\infty$ will have reached stationarity at time $t = -T$. If a chain with an unknown value $\mathbf{X}_{-T}$ is continued to run until $t = 0$, it will attain a value $\mathbf{X}_0 = \varphi^T(\mathbf{X}_{-T})$, which also comes from the stationary distribution. The CFTP algorithm searches for a time $-T$ such that the composite function $\varphi^T(\mathbf{X}_{-T})$ has a unique image *for all* arguments $\mathbf{X}_{-T}$. This implies that the chain started at $-T$ is equivalent to a chain started from $t = -\infty$, since it will reach the same state $\mathbf{X}_0$ regardless of its value at $t = -T$. Hence the sample $\mathbf{X}_0$ comes from the stationary distribution. Starting from the past and running into the present might seem counterintuitive and unnecessarily complicated. However, it is key for the algorithm to work and it can be shown that starting at $t = 0$ and coupling into the future will not guarantee that the samples are unbiased.

For large state spaces it is infeasible to monitor all initial conditions at time $-T$. However, this can be done efficiently if one can find a partial ordering over the state space that is preserved by the transition rule [12]:

$$\mathbf{X}_t \succeq \mathbf{Y}_t \quad \Rightarrow \quad \mathbf{X}_s \succeq \mathbf{Y}_s, \quad \forall s \geq t,$$

where $\succeq$ denotes the partial order, i.e., a binary relation which is reflexive, anti-symmetric and transitive, although it does not necessarily satisfy total comparability. Under these conditions, the whole state space can be monitored by checking for the coalescence of coupled Markov chains started at the upper and lower extremes of the state space [11,16].

Two Markov chains are said to be coupled if they use the same sequence of random numbers and the same transition rule but are started from different initial conditions. Coupled chains that meet at a time T_c are said to coalesce and will have identical states for $t > T_c$. A necessary (but not sufficient) condition for the preservation of the partial ordering is that the transition function is either monotone or anti-monotone:

$$\mathit{Monotonic}: \mathbf{X}_t \succeq \mathbf{Y}_t \Rightarrow \phi(\mathbf{X}_t) \succeq \phi(\mathbf{Y}_t)$$

$$\mathit{Anti\text{-}Monotonic}: \mathbf{X}_t \succeq \mathbf{Y}_t \Rightarrow \phi(\mathbf{X}_t) \preceq \phi(\mathbf{Y}_t),$$

for coupled chains $\mathbf{X}$ and $\mathbf{Y}$. If the partial order is preserved, we can monitor only the paths started at the 'extremes' of the state space, since all the paths in between remain bounded by them. We therefore define *upper* and a *lower* coupled Markov chains that enclose all other paths:

$$\begin{aligned}&\text{Lower path } (\mathbf{L}_t^{-T}) \text{ started from} \quad \mathbf{L}_{-T} = \mathbf{0}\\ &\text{Upper path } (\mathbf{U}_t^{-T}) \text{ started from} \quad \mathbf{U}_{-T} = \hat{\mathbf{1}},\end{aligned}$$

where $\hat{\mathbf{0}} \preceq \mathbf{x} \preceq \hat{\mathbf{1}}$, $\forall \mathbf{x}$.

The preservation of the partial order implies two important properties for coupled chains:

Sandwiching: all paths started between **L** and **U** will have coalesced by the time **L** and **U** do,

$$\mathbf{L}_t^{-T} \preceq \mathbf{X}_t^{-T} \preceq \mathbf{U}_t^{-T}, \quad t > -T. \tag{1}$$

Funneling: all paths will get closer if they are started further back into the past,

$$\begin{aligned}\mathbf{L}_t^{-T} \preceq \mathbf{L}_t^{-T-S} \preceq \mathbf{U}_t^{-T-S} \preceq \mathbf{U}_t^{-T},&\\ t > -T, S > 0.&\end{aligned} \tag{2}$$

If the state space is unbounded from above, we need to use Kendall's DCFTP construction. DCFTP works by introducing a time-evolving dominating process **D** with known stationary distribution, which provides a random upper bound to the state space. The original process **X** can then be generated as an adapted functional $\mathcal{F}$ of the dominating process and a *mark process* **M**:

$$\mathbf{D}_t^{-T} \succeq \mathbf{X}_t^{-T} = \mathcal{F}(\mathbf{D}_s, \mathbf{M}_s), -T \le s \le t. \tag{3}$$

The mark process generates a uniform random number each time **D** is changed. These marks are used to update the original process **X** according to the adapted functional (3) in a process that is equivalent to the direct simulation of **X** [12]. Heuristically, the DCFTP scheme works as follows. Since the dominating process is started from the stationary distribution at *t* = -*T*, D_t^{-T} is equivalent to a process started from *t* = -∞. By the funneling property, all chains from the original process started from *t* = -∞ will be beneath the dominating process: $\mathbf{X}_{-T}^{-\infty} \preceq \mathbf{D}_{-T}^{-\infty}$. If we set $\mathbf{U}_{-T} = \mathbf{D}_{-T}$ and $\mathbf{L}_{-T} = \hat{\mathbf{0}}$ and check that these two extreme paths coalesce, then all chains started from *t* = -*T* map to the same state at *t* = 0, due to the sandwiching property. It then follows that X_0^{-T} is equivalent to $X_0^{-\infty}$ and the sample comes from the stationary distribution of **X**, due to the equivalence of the adapted functional and the original process. Note that if **D** can be chosen to be a constant process equal to the maximal element of the state space, we obtain the CFTP algorithm [13].

These results are summarized in the following theorem for general DCFTP algorithms [12,13]:

Theorem 1 (DCFTP) *Consider a stationary dominating process* **D**, *for which* $\hat{\mathbf{0}}$ *is an ergodic atom, and an associated random mark process* **M**. *Suppose that the processes* $\mathbf{L} \preceq \mathbf{X} \preceq \mathbf{U}$ *are produced from* **D** *and* **M** *according to the adapted functional (3) so that the sandwiching and funneling properties (1)–(2) are fulfilled. Suppose further that* **X** *converges weakly to an invariant distribution* π *as* $t \to \infty$. *Then* **L** *and* **U** *will coalesce almost surely in finite time and, if coalescence is achieved,* $\mathbf{L}_0 = \mathbf{U}_0$ *is a sample from the stationary distribution* π.

Proof See [13].

Stochastic Simulation Algorithm (SSA)

This section presents briefly the classic Gillespie algorithm (SSA) for the exact simulation of the Master Equation of chemical reaction networks [7].

Definition 2 (Chemical reaction network) *A system of chemical reactions* $\mathcal{N}$ *is fully specified by the tuple* $\mathcal{N} = \{\mathcal{S}, \mathcal{R}, \Phi, v\}$, *where* $\mathcal{S} = \{S_1,\ldots,S_m\}$ *is a set of m different molecular species interacting through r reaction channels* $\mathcal{R} = \{R_1,\ldots,R_r\}$. *Each reaction* R_i *is described by a stoichiometry vector* v_i, *which gives the change in the number of molecules of all species when reaction* R_i *occurs, and a propensity function* $\Phi_i(\mathbf{x})$, *which gives the state-dependent probability that reaction* R_i *occurs. The state of the system is given by* $\mathbf{X}_t \equiv \mathbf{x}(t) = (x_1(t),\ldots,x_m(t)) \in \mathbb{N}^m$, *where each component* $x_i(t)$ *indicates the number of molecules of* S_i *at time t.*

Under the assumption that the molecules are confined to a well-stirred volume and held at constant temperature, we can formulate a ME governing the evolution of the system [7]:

$$\frac{dP(\mathbf{x},t)}{dt} = \sum_{i=1}^{r} \left[\Phi_i(\mathbf{x} - v_i)P(\mathbf{x} - v_i; t) - \Phi_i(\mathbf{x})P(\mathbf{x}; t) \right] \tag{4}$$

The ME is a conservation equation for the probability distribution and the right hand side accounts for the rate of change of the probability of finding the system in state **x**.

A general procedure to obtain exact realizations of Markov processes first suggested by Doob [17] was applied to chemical reactions by Gillespie in his celebrated Stochastic Simulation Algorithm [7]:

Algorithm 3 (SSA) *Given a chemical reaction network* $\mathcal{N} = \{\mathcal{S}, \mathcal{R}, \Phi, v\}$*, as in Definition 2, with initial state* $\mathbf{X}_{t_0}$ *and stopping time* T_s*:*

```
$k \leftarrow 0$
loop
  $k \leftarrow k + 1$
  $V_k,\ V'_k \sim U(0, 1)$
  for $i = 1$ to $r$ do
    $\theta_i \leftarrow \sum_{j=1}^{i} \Phi_j(\mathbf{X}_{t_{k-1}})$
  end for
  $t_k \leftarrow t_{k-1} - (1/\theta_r) \log V_k$
  if $t_k > T_s$ then
    return $\mathbf{X}_{t_{k-1}}^{t_0}$
  else
    $\mathbf{X}_{t_k} \leftarrow \mathbf{X}_{t_{k-1}} + v_i, R_{t_k} \leftarrow R_i \,\Big|\, \frac{\theta_{i-1}}{\theta_r} < V'_k < \frac{\theta_i}{\theta_r}$
  end if
end loop
```

A run of the SSA uses the uniform random numbers V, V' to generate a random sequence of reactions $\mathfrak{R} = \{R_{t_1}, \ldots, R_{t_n}\}$, taking place at the random transition times $\{t_1, \ldots, t_n\}$ such that $t_n < T_s < t_{n+1}$. The path $\mathbf{X}_{T_s}^{t_0} \equiv \{\mathbf{X}_{t_0}, \mathbf{X}_{t_1}, \ldots, \mathbf{X}_{t_n}\}$ is an exact stochastic realization of Eq. (4). Note that the sequence of reactions $\mathfrak{R}$ uniquely determines $\mathbf{X}_{T_s}^{t_0}$. For convenience, we have committed a slight of abuse of notation when using real valued indices to denote the state X_{t_k} and reaction R_{t_k} taking place at time t_k.

Henceforth, we represent compactly the SSA Markov process implemented by Algorithm 3 as:

$$\mathbf{X}_{T_s}^{t_0} = \mathcal{G}_{\mathrm{SSA}}(\mathcal{N}, \mathbf{X}_{t_0}, T_s). \qquad (5)$$

For an arbitrary initial state $\mathbf{X}_{t_0}$, repeated runs of the SSA will produce convergent estimates (in the Monte Carlo sense) of the distribution $P(\mathbf{x}, t | \mathbf{X}_{t_0}, t_0)$, $\forall t \in [t_0, T_s]$ [8]. However, if one is interested in the stationary distribution π, running the SSA repeatedly from different initial conditions for a finite time T_s does not guarantee that $P(\mathbf{x}, T_s)$ will converge to π, unless the starting points $\mathbf{X}_{t_0}$ are themselves drawn from π. Our Perfect Sampling algorithm addresses this issue.

Proof of the DCFTP-SSA for a class of networks of biochemical reactions

Viewing the SSA as the Markov process described by (5), we have developed a specific DCFTP algorithm that provides guaranteed sampling from the stationary distribution of the corresponding chemical ME [15]. We now provide a rigorous proof and an explicit implementation of the DCFTP-SSA for an important class of biochemical reactions relevant in gene regulation.

Partial ordering

We use the Pareto dominance relation, frequently used in economics, which is defined componentwise:

Lemma 4 (Partial order) *Given* $\mathbf{x}, \mathbf{y} \in \mathbb{N}^m$*, the relation* $\mathbf{x} \succeq \mathbf{y}$ *if* $x_i \geq y_i$*,* $\forall i$ *is a partial order.*

Proof The proof follows trivially from the properties of natural numbers:

Reflexivity: $\forall x_i \in \mathbb{N}$, $x_i \geq x_i$, whence $\mathbf{x} \succeq \mathbf{x}$

Anti-symmetry: $\forall x_i, y_i \in \mathbb{N}$, if $x_i \geq y_i$ and $y_i \geq x_i$ then $x_i = y_i$. This means that $\mathbf{x} \succeq \mathbf{y}$ and $\mathbf{y} \succeq \mathbf{x}$ implies $\mathbf{x} = \mathbf{y}$

Transitivity: $\forall x_i, y_i, z_i \in \mathbb{N}$, if $x_i \geq y_i$ and $y_i \geq z_i$ then $x_i \geq z_i$. And the same property applies to the vectors: $\mathbf{x} \succeq \mathbf{y}$ and $\mathbf{y} \succeq \mathbf{z}$ implies $\mathbf{x} \succeq \mathbf{z}$. □

Assumptions on the reaction network

Consider a system of chemical reactions as given by Definition 2 with state vector $\mathbf{x}(t) \in \mathbb{N}^m$. To guarantee the preservation of the Pareto partial order under the SSA Markov process (5), we restrict ourselves to a class of chemical networks with the following properties:

(a) all reactions are *uni-molecular birth-death* processes with non-zero propensities, i.e., each reaction R_i will only modify one species S_j by adding or subtracting one molecule. The reactions can be divided into two subsets:

$$\begin{array}{lll}\text{Birth :} & & \\ \mathcal{R}^+ & = & \{R_i \mid \nu_i = (0,\ldots,0,+1,0,\ldots,0) \\ & \Rightarrow & \mathbf{x}+\nu_i \geq \mathbf{x}\}, \\ \text{Death :} & & \\ \mathcal{R}^- & = & \{R_i \mid \nu_i = (0,\ldots,0,-1,0,\ldots,0) \\ & \Rightarrow & \mathbf{x}+\nu_i \leq \mathbf{x}\}\end{array}$$

(b) the system must be *chemically reversible*, i.e., every reaction must be reversible leading to an irreducible Markov process

(c) all death reactions must be linear, i.e.

$$\Phi_i = k_j x_j \quad \text{for} \quad R_i \equiv X_j \to \varnothing$$

(d) all birth reactions must have *(anti-)monotonic, sub-linear propensity functions*, i.e., $\forall i$, j, $\forall \mathbf{x}$: $\partial\Phi_i(\mathbf{x})/\partial x_j$ does not change sign and Φ_i can be bounded by a linear function (or a constant).

As shown below, the last two assumptions are related to domination by a linear network which is required to have a stationary distribution.

Although assumptions (a) – (d) might appear restrictive, the specified class of reactions is generic and encompasses the standard equations used in the modelling of genetic and regulatory networks, the cellular circuits where stochasticity is most significant. Note that assumption (c) is not unrealistic for models of gene regulatory networks, in which linear death terms due to the cellular environment are prevalent. Birth reactions in these models are usually represented through *nonlinear*, uni-molecular (compound) rate laws that appear from quasi steady-state approximations. These functional forms have been shown to work well in the stochastic setting [18]. Our own simulations confirmed that they provide a good approximation in a wide range of parameters (results not shown). These compound rate laws are the key components that encode the positive and negative feedback in gene regulation. Classic examples are the sigmoid functions:

$$\text{Monod positive feedback :} \quad m(x) = \frac{kx^{\alpha}}{\theta^{\alpha}+x^{\alpha}}, \qquad (6)$$

$$\text{Hill negative feedback :} \quad h(x) = \frac{k\theta^{\alpha}}{\theta^{\alpha}+x^{\alpha}}, \qquad (7)$$

which are sub-linear, (anti-)monotonic functions.

Dominating process and adapted functionals

As stated above, assumption (d) is related to domination. In general, the state space of chemical reaction networks is unbounded from above; hence we must use the DCFTP construction, which requires a dominating process $\mathbf{D}$ with known stationary distribution. Fortunately, it has been shown that any network of *linear* first order reactions has a stationary distribution which is multivariate Poisson [19]. Moreover, it can be shown that $\hat{\mathbf{0}}$ is an ergodic atom for the multivariate Poisson, as assumed in Theorem 1 [13]. It then follows that a dominating process for any reaction network $\mathcal{N} = \{\mathcal{S}, \mathcal{R}, \Phi, \nu\}$ composed of uni-molecular, sub-linear, (anti-)monotonic birth-death processes, as defined above, can be found by 'linearizing' the original network; that is, by constructing a linearized version of this network $\tilde{\mathcal{N}} = \{\mathcal{S}, \mathcal{R}, \tilde{\Phi}, \nu\}$, with the same reactions and compounds but with linear propensities $\tilde{\Phi}_i(\mathbf{x}) \geq \Phi_i(\mathbf{x})$, $\forall \mathbf{x}$, $\forall i$ that bound the original Φ from above. Under conditions of stability, the ME of $\tilde{\mathcal{N}}$ will have a stationary distribution $\tilde{\pi}$, given by a multivariate Poisson that can be obtained by solving a system of linear equations [19]. The existence of the stationary distribution of the dominating linear network $\tilde{\mathcal{N}}$ guarantees the existence of the stationary distribution for the original network of reversible, uni-molecular nonlinear reactions $\mathcal{N}$.

The *dominating process* $\mathbf{D}$ is defined as the stationary SSA process (5) of the linearized network $\tilde{\mathcal{N}}$ with initial state sampled from $\tilde{\pi}$:

$$\mathbf{D}_T^{t_0} = \mathcal{G}_{\text{SSA}}(\tilde{\mathcal{N}}, \mathbf{D}_{t_0}, T), \quad \mathbf{D}_{t_0} \sim \tilde{\pi}, \qquad (8)$$

with the sequence of reactions $\tilde{\mathfrak{R}} = \{\tilde{R}_{t_1}, \ldots\}$.

It has been shown [13] that a correct realization of the original (nonlinear) SSA process $\mathbf{X}$ for a network $\mathcal{N}$ with monotonic propensities can also be obtained through an *adapted functional* $\mathcal{F}$ defined in terms of the dominating process $\mathbf{D}$ and a random mark process $\mathfrak{M} = \{M_{t_1}, \ldots\}$ where $M_{t_k} \sim U(0, 1)$:

$$\mathbf{X}_T^{t_0} = \mathcal{F}(\mathbf{D}_T^{t_0}, \mathfrak{M}).$$

The update rule for $\mathcal{F}$ uses the ratio of the monotonic propensity functions of the original and dominating processes as follows:

$$\begin{aligned}\mathbf{X}_{t_k} &= \phi_{\mathcal{F}}(\mathbf{X}_{t_{k-1}}, \mathbf{D}_{t_{k-1}}, M_{t_k}) \\ &= \begin{cases} \mathbf{X}_{t_{k-1}} + \nu_{t_k} & \text{if } M_{t_k} < \Psi_{t_k}(\mathbf{X}_{t_{k-1}}, \mathbf{D}_{t_{k-1}}) \\ \mathbf{X}_{t_{k-1}} & \text{otherwise,} \end{cases}\end{aligned} \qquad (9)$$

where $\Psi_{t_k}(\mathbf{X}_{t_{k-1}}, \mathbf{D}_{t_{k-1}}) \equiv \Phi_{t_k}(\mathbf{X}_{t_{k-1}}) / \tilde{\Phi}_{t_k}(\mathbf{D}_{t_{k-1}})$ and ν_{t_k}, Φ_{t_k}, $\tilde{\Phi}_{t_k}$ correspond to reaction $\tilde{R}_{t_k}$ in the reaction sequence $\tilde{\mathfrak{R}}$.

The necessary ingredient for the DCFTP is the construction of an order-preserving Markov process for the evolution of two chains **X** and **Y** coupled to the dominating process **D**. For our network $\mathcal{N}$, this process is defined as:

$$(\mathbf{X}_T^{t_0}, \mathbf{Y}_T^{t_0}) = \mathcal{F}(\mathbf{D}_T^{t_0}, \mathfrak{M}), \qquad (10)$$

with transition rule:

$$\begin{aligned}(\mathbf{X}_{t_k}, \mathbf{Y}_{t_k}) &= \phi_{\mathcal{F}}(\mathbf{X}_{t_{k-1}}, \mathbf{Y}_{t_{k-1}}, \mathbf{D}_{t_{k-1}}, M_{t_k}) \\ &= \begin{cases} (\phi_{\mathcal{F}}(\mathbf{X}_{t_{k-1}}, \mathbf{D}_{t_{k-1}}, M_{t_k}), \phi_{\mathcal{F}}(\mathbf{Y}_{t_{k-1}}, \mathbf{D}_{t_{k-1}}, M_{t_k})) \\ \text{if } \tilde{R}_{t_k} \text{ is monotone} \\ (\phi_{\mathcal{F}}(\mathbf{Y}_{t_{k-1}}, \mathbf{D}_{t_{k-1}}, M_{t_k}), \phi_{\mathcal{F}}(\mathbf{X}_{t_{k-1}}, \mathbf{D}_{t_{k-1}}, M_{t_k})) \\ \text{if } \tilde{R}_{t_k} \text{ is anti-monotone,} \end{cases}\end{aligned} \qquad (11)$$

where the componentwise transition rule is given in Eq. (9). The transition rule $\phi_{\hat{\mathcal{F}}}$ incorporates the cross-over scheme in which the processes **X** and **Y** use the state of each other when determining their update, as introduced by Häggström and Nelander to deal with anti-monotonic processes [20].

Proof

We now show that the partial ordering defined in Lemma 4 is indeed preserved under the evolution given by Eqs. (10)–(11) for the class of reactions specified above.

Lemma 5 (Preservation of partial ordering) *Consider a chemically reversible reaction network* $\mathcal{N}$ *of uni-molecular, sub-linear, (anti-)monotone birth-death reactions and its associated SSA dominating process* **D**, *obtained from the linearized network* $\tilde{\mathcal{N}}$, *with the sequence of events* $\tilde{\mathfrak{R}} = \{\tilde{R}_{t_1}, \ldots\}$. *Consider two coupled chains* **X** *and* **Y** *evolving under (10)–(11), where* $\mathfrak{M} = \{M_{t_1}, \ldots\}$ *is a sequence of random marks. Then* $\mathbf{X}_t \succeq \mathbf{Y}_t \Rightarrow \mathbf{X}_s \succeq \mathbf{Y}_s$, $\forall s > t$.

Proof Assume $\mathbf{X}_{t_0} \succeq \mathbf{Y}_{t_0}$ throughout. First consider the case when $\tilde{R}_{t_1}$ is monotonic. Then the possible outcomes for $t_0 < s < t_1$ are:

$$\begin{aligned} &(i) && \Psi_{t_1}(\mathbf{Y}_{t_0}, \mathbf{D}_{t_0}) \le \Psi_{t_1}(\mathbf{X}_{t_0}, \mathbf{D}_{t_0}) \le M_{t_1} \\ &(ii) && \Psi_{t_1}(\mathbf{Y}_{t_0}, \mathbf{D}_{t_0}) \le M_{t_1} < \Psi_{t_1}(\mathbf{X}_{t_0}, \mathbf{D}_{t_0}) \\ &(iii) && M_{t_1} < \Psi_{t_1}(\mathbf{Y}_{t_0}, \mathbf{D}_{t_0}) \le \Psi_{t_1}(\mathbf{X}_{t_0}, \mathbf{D}_{t_0}). \end{aligned}$$

Outcome (i) means that neither **X** nor **Y** is modified and the preservation of the partial order is obvious. For (iii), both are modified by the same amount ν_{t_1} and the order is preserved. The interesting case is (ii) for which **X** is modified but not **Y**. If $\tilde{R}_{t_1} \in \mathcal{R}^+$, then $\mathbf{X}_{t_1} \succeq \mathbf{X}_{t_0} \succeq \mathbf{Y}_{t_1} = \mathbf{Y}_{t_0}$ which also implies order preservation. However, if $\tilde{R}_{t_1} \in \mathcal{R}^-$, then it is possible for the two chains to coalesce if $\mathbf{X}_{t_0} + \nu_{t_1} = \mathbf{Y}_{t_0}$. Note that since, by uni-molecularity, only unit changes of the states are allowed, it is impossible for two paths to cross.

When $\tilde{R}_{t_1}$ is anti-monotone, the outcomes are:

$$\begin{aligned} &(iv) && \Psi_{t_1}(\mathbf{X}_{t_0}, \mathbf{D}_{t_0}) \le \Psi_{t_1}(\mathbf{Y}_{t_0}, \mathbf{D}_{t_0}) \le M_{t_1} \\ &(v) && \Psi_{t_1}(\mathbf{X}_{t_0}, \mathbf{D}_{t_0}) \le M_{t_1} < \Psi_{t_1}(\mathbf{Y}_{t_0}, \mathbf{D}_{t_0}) \\ &(vi) && M_{t_1} < \Psi_{t_1}(\mathbf{X}_{t_0}, \mathbf{D}_{t_0}) \le \Psi_{t_1}(\mathbf{Y}_{t_0}, \mathbf{D}_{t_0}). \end{aligned}$$

As above, outcomes (iv) and (vi) lead to no change in relative order. For (v), again we update **X** but not **Y** due to the crossover scheme. As for the monotone case, if $\tilde{R}_{t_1} \in \mathcal{R}^+$ this leads to order preservation, while if $\tilde{R}_{t_1} \in \mathcal{R}^-$ it is possible for the two chains to coalesce.

It thus follows that **X** and **Y** maintain their partial ordering through every update of the (anti-)monotone process. The proof for $s > t_1$ follows by induction. □

Note that the dominated processes given by Eqs. (9) and (11) become identical when **X** = **Y**. Therefore, after coalescence the dominated process is statistically identical to the original SSA process. Since we have found a dominating process and an adapted functional, we can use Theorem 1 to obtain:

Theorem 6 (DCFTP-SSA) *Under the assumptions of Lemma 5, Theorem 1 is fulfilled and the DCFTP-SSA described in Algorithm 7 will produce a sample from the stationary distribution of the original process* **X** *with a coalescence time which will be finite almost surely.*

Proof The sandwiching (1) and funneling (2) properties follow from the preservation of partial ordering (Lemma 5) [12]. The remainder can be adapted from the general Theorem 1. □

Algorithm

A brief outline of the DCFTP-SSA is as follows:

Algorithm 7 (DCFTP-SSA) *Given a reversible system of unimolecular birth-death chemical reactions $\mathcal{N}$ with (anti-)monotone, sub-linear propensity functions, obtain its linearized version $\tilde{\mathcal{N}}$ with multivariate Poisson stationary distribution $\tilde{\pi}$:*

```
$T \leftarrow 1$
$\breve{\mathbf{D}}_0^0 \sim \tilde{\pi}, \breve{\mathbf{M}}_0^0 \sim U(0,1)$
$\breve{\mathbf{D}}_T^0 \leftarrow \mathrm{Extend}(\breve{\mathbf{D}}_0^0, T)$
$\breve{\mathbf{M}}_T^0 \leftarrow \mathrm{GenerateMarks}(\breve{\mathbf{M}}_0^0, T)$
loop
  $(\mathbf{D}_0^{-T}, \mathbf{M}_0^{-T}) \leftarrow \mathrm{Reverse}(\breve{\mathbf{D}}_T^0, \breve{\mathbf{M}}_T^0)$
  $(\mathbf{L}_0^{-T}, \mathbf{U}_0^{-T}) \leftarrow \mathrm{Evolve}(\mathbf{D}_0^{-T}, \mathbf{M}_0^{-T})$
  if $\mathbf{U}_0 = \mathbf{L}_0$ then
    return $\mathbf{L}_0$
  end if
  $T \leftarrow 2T$
  $\breve{\mathbf{D}}_T^0 \leftarrow \mathrm{Extend}(\breve{\mathbf{D}}_{T/2}^0, T/2)$
  $\breve{\mathbf{M}}_T^0 \leftarrow \mathrm{GenerateMarks}(\breve{\mathbf{M}}_{T/2}^0, T/2)$
end loop
```

The function Extend($\breve{\mathbf{D}}_{T/2}^0$, $T/2$) runs Algorithm 3 for the linearized network $\tilde{\mathcal{N}}$ and appends the path $\mathcal{G}_{\mathrm{SSA}}(\tilde{\mathcal{N}}, \breve{\mathbf{D}}_{T/2}, T/2)$ to the path $\breve{\mathbf{D}}_{T/2}^0$. Similarly, the function GenerateMarks appends paths generated from a uniform distribution to extend the mark process. Both the marks and the forward dominating path are then reversed in time by the function Reverse. Extending these processes *backwards* in time in this manner is justified because of their stationarity and reversibility, which allows us to reverse the processes and translate them in time [9]. Finally, the Evolve function starts the coupled upper and lower chains from $\mathbf{L}_{-T} = \hat{\mathbf{0}}$ and $\mathbf{U}_{-T} = \mathbf{D}_{-T}$ and evolves them forward as described by Eq. (11). Note that the assumption of reversibility of the network ensures that the reverse process will be forward-evolvable. Our requirement that propensities are non-zero also ensures that reactions are not eliminated from the network. If this were to happen, it would effectively make the system irreversible. If **L** and **U** have not coalesced at $t = 0$, **D** and **M** are extended further back in time and **L** and **U** are restarted. Doubling the starting time at each iteration has been shown to be reasonably efficient (see [11] for a discussion).

Applications of the algorithm

Numerical convergence: First order reaction

To characterize numerically the convergence properties of the DCFTP-SSA, consider the first order reaction where species A is created at a (normalized) constant rate k from a source and degraded to a sink:

$$\begin{aligned} &\varnothing \xrightarrow{k} A \xrightarrow{1} \varnothing \\ \dot{P}_j &= kP_{j-1} - kP_j + (j+1)P_{j+1} - jP_j \\ &\equiv (\mathbb{E}^{-1} - 1)kP_j + (\mathbb{E} - 1)jP_j. \end{aligned} \tag{12}$$

Here P_j denotes the probability of having j molecules of A and $\mathbb{E}$ and $\mathbb{E}^{-1}$ are step operators [4]:

$\mathbb{E} f(j) = f(j+1)$ and $\mathbb{E}^{-1} f(j) = f(j-1)$ acting on a function $f(j)$. For the usual initial condition with 0 molecules, the time-dependent solution of Eq. (12) is a Poisson distribution with time-dependent parameter $k(1 - e^{-t})$ [15]. Equation (12) is an instance of the immigration-death process which appears in different settings in the stochastic processes literature.

If, as a proxy for sampling the stationary distribution π, one obtains samples of $P(j, T_s|0, 0)$ from repeated runs of

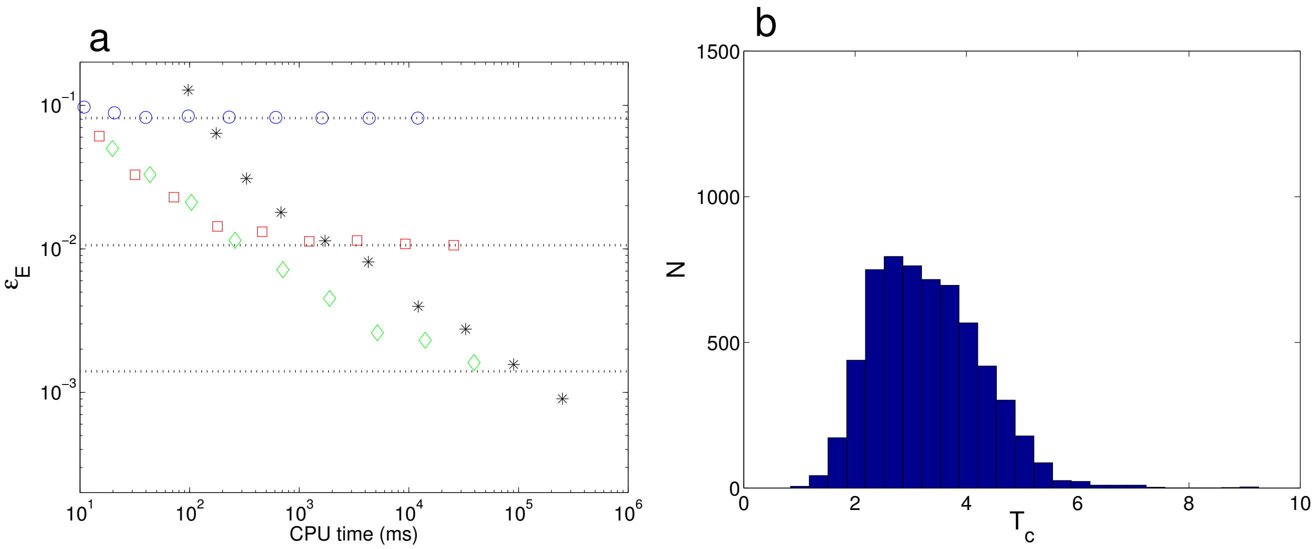

Figure 1

Convergence of the DCFTP-SSA for the first order reaction (12). (*a*) As a function of CPU time, we represent the Euclidean error $\in_E$ of the stationary distribution of Eq. (12) with *k* = 5 sampled with the DCFTP-SSA (+) and with the standard SSA with stopping times T_s = 2(❍), 4(□), 6(◊). For this simple ME, the limiting value of the Euclidean error of the finite-time SSA is $\dagger_E(T_s)^2 = \sum_{j=1}^{\infty}(\pi_j - P_j(T_s))^2 = I_0(2k)e^{-2k} - 2I_0(2k\sqrt{\alpha})e^{-k-\alpha} + I_0(2k\alpha)e^{-2\alpha}$, where α = 1 - exp(-T_s) and $I_0(x)$ is the modified Bessel function of the first kind [15]. This means that SSA simulations that are run for a time T_s will converge to a systematic sampling error, indicated by the dotted lines. This source of error is eliminated when using the DCFTP-SSA, which shows no flooring for $\in_E$ and the expected $N^{-1/2}$ scaling with the number of Monte Carlo samples [26]. The guarantees provided by the DCFTP-SSA come at a modest computational cost, which is comparable to that of long SSA runs. (*b*) The distribution of coalescence times T_c for the DCFTP-SSA is relatively symmetric and concentrated around the mean with a rapid decay for long times. The data presented corresponds to 6000 runs. This distribution reflects the benign structure of the unimodal stationary distribution of this particular ME, which makes long coalescence times unlikely.

the SSA for a finite time T_s, this will lead to a systematic error that will not disappear as the number of samples (and the CPU time) is increased. The use of the DCFTP-SSA eliminates this source of error, as shown in Fig. 1a. This figure also shows that the guaranteed convergence of the DCFTP-SSA incurs a modest additional CPU cost. The increased computational cost is twofold: increased memory requirements, since we need to store the history of the dominating process as well as the sequence of random numbers used to update the coupled chains; and longer running times, since we need to extend the process backwards for an indefinite (unbounded) amount of time. Fig. 1b presents the statistics of the coalescence times for this reaction. In this simple reaction, the distribution of stopping times is relatively symmetric and concentrated around the mean value, without the long tails that would correspond to long runs started a long time into the past. As the next example shows, the distribution of coalescence times reflects the complexity of the structure of the stationary distribution.

Multistability: Genetic toggle switch

The mutual activation and repression of groups of genes in regulatory networks can lead to multi-stability allowing cells to attain different states [5,21]. An important and difficult problem is to find the probabilities of the different states and the expected switching times. Previously [15], we applied the DCFTP-SSA to the standard toggle switch with two Hill-repressed genes [22]. We now apply the algorithm to a more complex model of two mutually activating genes [21] with a complicated activation function which is not of the standard Monod form:

$$\begin{tikzcd}
\varnothing \arrow[r, "f(n_B)"] & A \arrow[r, "1"] \arrow[d, phantom, "\times"] & \varnothing \\
\varnothing \arrow[r, "f(n_A)"'] & B \arrow[r, "1"'] & \varnothing
\end{tikzcd}$$

$$\begin{aligned}\dot{P}_{n_A,n_B} = {} & (\mathbb{E}_A^{-1}-1)f(n_B)P_{n_A,n_B} \\ & +(\mathbb{E}_A-1)n_A P_{n_A,n_B} \\ & +(\mathbb{E}_B^{-1}-1)f(n_A)P_{n_A,n_B} \\ & +(\mathbb{E}_B-1)n_B P_{n_A,n_B},\end{aligned} \tag{13}$$

with the activation given by

$$f(n_i) = \gamma + \frac{\kappa_i n_i^4}{\kappa_{i0}+\kappa_{i1}n_i+\kappa_{i2}n_i^2+\kappa_{i3}n_i^3+n_i^4},$$

where n_A and n_B are the number of protein molecules, $\gamma$is the basal production rate and κ_{ij} are parameters. The functional form of the activation appears as a consequence of particular properties of this system: each transcription site can be occupied by up to four monomers and becomes activated when a tetramer is bound. However, note that $f(n)$ is monotonic and sub-linear and therefore the DCFTP-SSA is applicable.

For certain choices of parameters, the stationary distribution of the system is bimodal: the peak located at the origin corresponds to both genes being 'off', while the other mode indicates both genes are 'on' (Fig. 2a). The extreme bimodality of this distribution makes its sampling difficult using the standard SSA. As can be seen in Fig. 3a, if we start from the initial condition (0, 0), the standard SSA levels off in a similar manner to Fig. 1a, highlighting the presence of a systematic error. In contrast, the DCFTP-SSA converges to the stationary distribution at the expected $N^{-1/2}$ rate.

Figure 2b also shows that the probability sampled with the DCFTP-SSA captures the global structure of the probability distribution even in this extreme example. On the other hand, closer inspection of the SSA simulations started from the (0, 0) reveals that for short stopping times, the process remains at the mode located near the origin (Fig 2c). Although simple heuristics on how to choose the initial condition have been suggested to improve the sampling of π with the SSA, Figure 2 shows that similar mis-sampling errors appear if we run the standard SSA from a variety of initial conditions. Fig. 2d shows that sampling the initial condition from a uniform grid in state space does not capture the full features of the distribution since this initial condition does not represent a consistent sampling for stationarity. If we use the fixed points of the corresponding deterministic system as initial conditions for the SSA, we would still lack the probability mass associated with each mode. For instance, starting half of the simulations at the origin and the remaining at the other fixed point provides little improvement since almost half of the simulations remain near the origin (Fig. 2e). Similar errors appear if we sample a long SSA run at fixed intervals Δt to provide independent samples as intial conditions (Fig. 2f), or even if we use samples drawn from the true stationary distribution as initial conditions for the SSA.

We can understand why the stationary distribution of this system presents such a challenge for the finite-time SSA by considering the mean first passage times. Figure 4a shows the average time to reach all other states from the origin and Fig. 4b shows the average time to reach the other mode. To be certain that an SSA run will produce correct samples from the stationary distribution, it must visit each mode several times. For the system considered in Fig. 2 we need stopping times on the order of 10^7 to be certain that the simulation has not been stuck in one mode. With our implementation, the DCFTP compares favorably with the SSA wtih $T_s = 10^7$ (data not shown).

Figure 3a summarizes the CPU times for the different SSA sampling schemes shown in Fig. 2 compared to the DCFTP-SSA. Again, the DCFTP-SSA introduces a reasonable overhead but provides guarantees that no systematic sampling error exists. To understand how the extreme bimodality of this distribution affects the running time of the DCFTP-SSA, Figure 3b shows the statistics of the coalescence times for this system. As compared with Fig. 1b, the distribution of coalescence times is bimodal with a second mode at long coalescence times a long tail. This reflects the complex structure of the stationary distribution in state space which induces longer coalescence times to guarantee the correct sampling. As explained in the Discussion section, the numerical performance of the algorithm in situations where long runs are more likely can be improved by the use of rejection sampling schemes.

This simple example illustrates the potential pitfalls of using the standard SSA for multimodal systems with long switching time-scales. If the SSA is run with too short stopping times, one runs the risk of missing important features of the distribution that could lead to erroneous conclusions about the number and relative weight of possible states. These problems become more acute as the dimensionality of the state space increases.

Steady-state dynamics: Generalized repressilator

Although regularity and robustness are important for their reliable operation in time-keeping, circadian and synchronization processes, cellular oscillators have a biochemical basis and are subject to high levels of noise [23,24]. In previous work [15], we studied the stochastic version of the repressilator, a synthetic transcriptional oscillator that consists of three mutually repressing genes in a loop (Fig. 5a) and has been implemented in *E. coli* [23]. Experi-

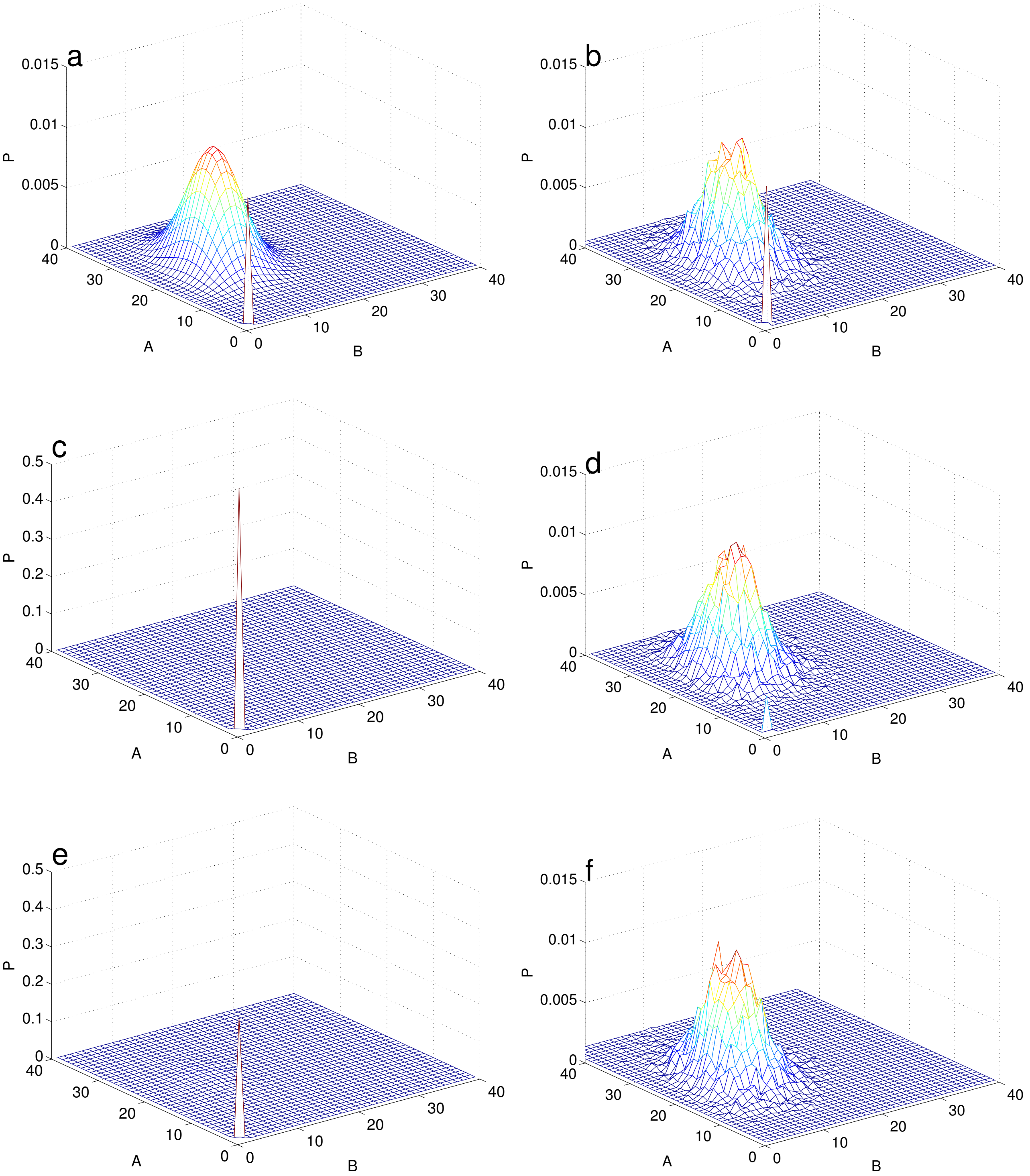

Figure 2 *(caption on next page)*

Figure 2 *(see previous page)*
Sampling of the stationary distribution for the bistable gene network (13) using different methods. (*a*) The 'true' stationary probability distribution π for the ME (13) calculated numerically with the approximate eigenvector method [15]. The parameters are $\kappa_B = 25$, $\kappa_A = 12$, $\kappa_{A0} = \kappa_{B0} = 60$, $\kappa_{A1} = \kappa_{B1} = 10$, $\kappa_{A2} = \kappa_{B2} = \kappa_{A3} = \kappa_{B3} = 1$, and $\gamma = 0.01$. The locations of the two modes match the fixed points of the corresponding deterministic system. Note the extreme asymmetry of the bimodal probability distribution. (*b*) The estimate of π obtained from 10^4 samples of the DCFTP-SSA reproduces the presence of both modes and their relative weights. (*c*) Estimate of π from 10^4 samples of the SSA started at (0,0) with $T_s = 10^3$. (*d*) Estimate of π obtained from 10^4 SSA simulations started from 10^4 different initial conditions chosen uniformly at random on the 100 × 100 lattice closest to the origin and run for $T_s = 10^3$. (e) Estimate of π obtained from 10^4 SSA simulations, 5000 of them started from the origin and the other 5000 from the other mode and run for $T_s = 10^3$. (*f*) Estimate of π obtained from 10^4 samples from a long SSA run sampled at interval $\Delta t = 10^3$. Note the different scale on the z-axis for (*c*) and (e) and how the SSA runs (*c*)-(*f*) do not capture the overall structure of π.

ments on the original repressilator showed that the oscillations are very noisy and stochastic models are required to capture these features.

Here, we investigate the stochastic properties of the generalized repressilator with an arbitrary number *n* of genes in the loop [25]. Müller *et al* studied the deterministic version and showed that the system oscillates when *n* is odd, as expected by analogy with the ring oscillator in electronic circuits (see Fig. 5b). This system allows us to study the dependence of the variability of the oscillations with the number of genes and to showcase the scalibility of our algorithm as the number of variables (and the dimensionality of the state space) increases.

We now use the DCFTP-SSA to characterize the periodicity of the stochastic oscillations of the generalized repressilator:

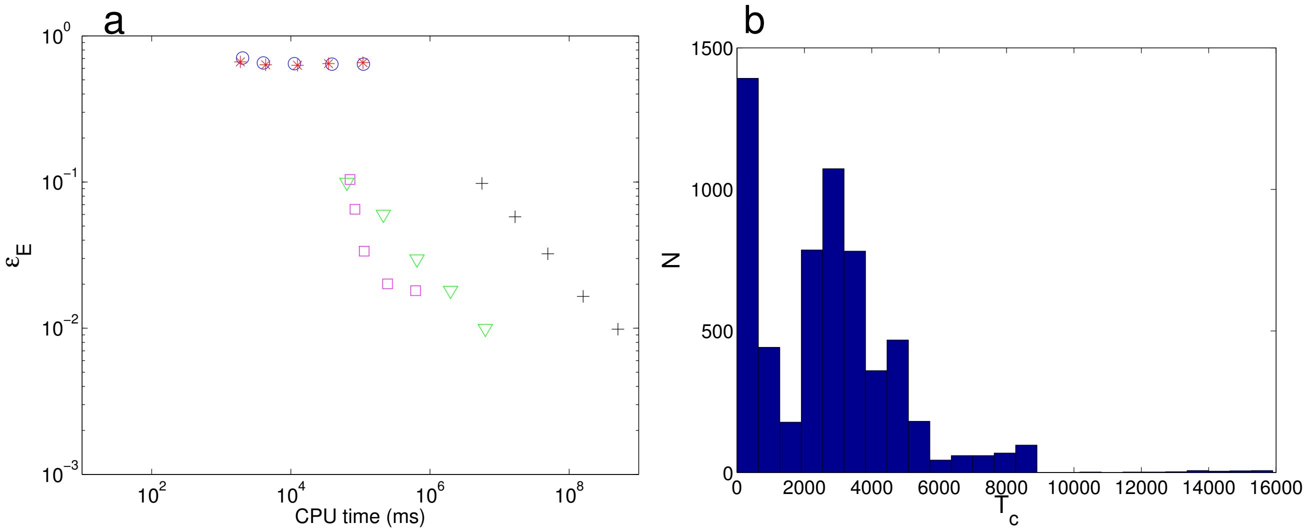

Figure 3
Convergence of the DCFTP-SSA for the bistable gene network (13). (*a*) As a function of CPU time, we represent $\in_E$, the Euclidean error of the sampled distributions estimated using: the DCFTP-SSA (+), as in Fig. 2 (*b*); the SSA with T_s = 1000(❍), as in Fig. 2 (*c*); the SSA started from the two modes (*), as in Fig. 2 (*d*); the SSA started from uniform initial conditions (∇), as in Fig. 2 (e); and the SSA uniformly sampled from a long run (□), as in Fig. 2 (*f*). For each scheme, we produced *N* = 100, 316, 1000, 3162 and 10000 samples to show how the error improves as the number of samples increases. The DCFTP-SSA converges to the stationary distribution at the expected $N^{-1/2}$ rate, whereas the approximate estimates obtained using the SSA level off in a similar manner as in Fig. 1*a*. (*b*) The distribution of coalescence times for the DCFTP-SSA for this network is bimodal with a very long tail for the second mode, indicating the likelihood of long coalescence times. The data presented corresponds to 6000 runs.

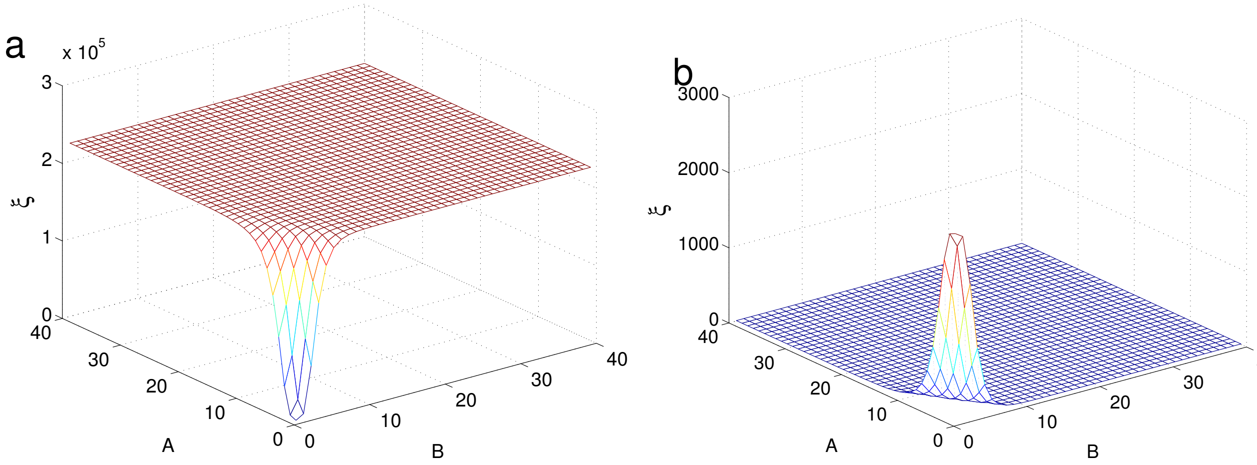

Figure 4
Mean transition times for the bistable gene network (13). (*a*) The mean first passage time ξ to reach the origin for the lattice points of the state space close to the origin. The escape time from the mode located away from the origin is 2 × 10^5. (*b*) The mean first passage time from the origin to the other mode is 3 × 10^3.

$$\begin{aligned}\dot{P}_j &= \sum_{i=0}^{n}(\mathbb{E}_{M_i}^{-1}-1)\frac{k_M}{(1+R_{i+1}^{\alpha})}P_j \\ &+ (\mathbb{E}_{M_i}-1)d_M M_i P_j \qquad (14)\\ &+ (\mathbb{E}_{R_i}^{-1}-1)(k_B+k_R M_i)P_j \\ &+ (\mathbb{E}_{R_i}-1)R_i P_j,\end{aligned}$$

where the shorthand P_j denotes the state $P_{M_0,\ldots,M_n,R_0,\ldots,R_n}$ and all integers are i mod n. Here M_i are the mRNA levels (with production rate k_M and degradation rate d_M) and R_i are the corresponding proteins (with basal rate k_B and linear production rate k_R). The repressilator network fulfills the conditions of applicability of the DCFTP-SSA and we have used our algorithm to generate time-series which are guaranteed to be at stationarity. The fact that the system has a persistent, oscillatory dynamics does not preclude it from being stationary. As expected, our DCFTP-SSA simulations show that the stationary distribution π conforms to the shape of a circular ridge in $2n$-dimensional space, which is directly related to the deterministic limit cycle [4]. In this case, the probability mass is unimodal along the ridge, which means that sampling from a long time-series is unproblematic since there is no risk of avoiding regions of state space that have high probability.

As one would expect from the deterministic analysis, the mean period increases linearly with n (Figure 5d). This follows from the fact that the oscillatory behavior propagates in a wave-like manner around the loop. If we assume that the period is formed as the sum of n independent genes rising and falling in sequence, then a circuit with n genes will have a period whose mean scales linearly with n, as shown in Figure 5d in accordance with the deterministic model. However, the shape and moments of the distribution of the periods change significantly as a function of n, as shown in Figs. 5c–d. The distribution of the period for shorter circuits will necessarily be right-skewed since there is a minimal waiting time, akin to a refractory period, before the gene can rise again. This asymmetry is observed in the case of n = 3 but has almost disappeared for n = 9, and is captured by the skewness, which decreases towards zero as n increases.

Our numerics also indicate that the relative variability of the period is not constant as the number of genes in the loop increases. Figure 5d shows that the variance of the period increases quadratically, which implies that the successive periods are not independent. This implies that, for the set of parameters in Figure 5, there is an optimal length of n^* = 7 genes in the loop, for which the relative fluctuations of the period, as measured by the coefficient of variation, are minimal. Note also that the kurtosis remains almost constant and positive, which indicates that there are fat tails even for longer circuits. Interestingly, the kurtosis also attains a shallow minimum at n^* = 7, indicating a relative decrease in the dispersion of the distribution. Another important characteristic of an oscillator is the *rise time*, which gives an indication of its precision. Our numerics find no change in the variance of the rise times as the number of genes increases (results not

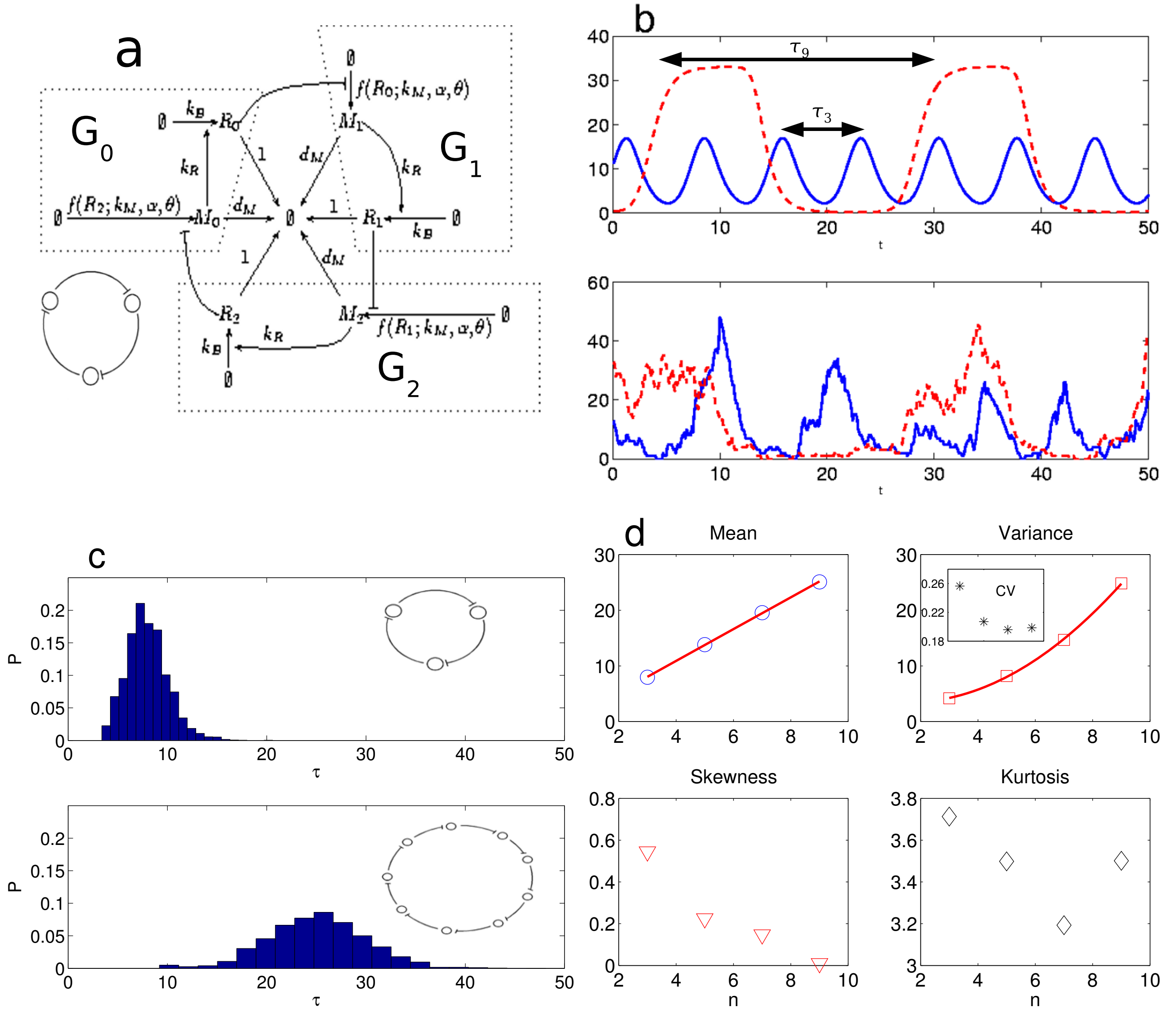

Figure 5

Noise characteristics of the generalized repressilator (14). (*a*) Detailed diagram of the reactions in the standard repressilator with three genes involving six chemical species, as implemented with our stochastic algorithm. In the simplified cartoon, each circle represents a gene repressing the subsequent gene in a cycle. The generalized repressilator studied here considers cycles with odd number of genes *n* = 3, 5, 7, 9. (*b*) The top panel shows time series of one of the proteins for the deterministic model of the repressilator with *n* = 3 (filled) and *n* = 9 (dashed) genes with parameters k_M = 25, d_M = 3, θ = 3, k_R = 4 and α = 2. The lower panel shows the corresponding time series of the SSA started from stationarity, guaranteed by the DCFTP-SSA. For the top panel, the y-axis has units of protein concentration, whereas for the lower panel the *y*-axis has unitos of number of proteins. (*c*) The top panel shows the distribution of the period for the repressilator with *n* = 3 genes, while the bottom panel shows the same distribution for the generalized repressilator with *n* = 9 genes. Note that the distribution for *n* = 3 is skewed with a long right tail, while that of *n* = 9 is more symmetric, but has fatter tails than would be expected for a Gaussian distribution. The histograms were obtained from time-series with 10^4 periods. (*d*) The top two panels show the dependence of the mean (◦) and variance (□) of the period distribution with *n*. The lines indicate a linear fit for the means and a quadratic fit for the variances. The inset in the top right panel, shows that, for this set of parameters, the relative noise of the period, as measured by the coefficient of variation (*), is minimal for a length of *n* = 7 genes in the loop. The two lower panels show the skewness (∇) and kurtosis (◊) for the period distribution. The skewness decreases to zero as *n* grows, in accordance with the observed decrease of the asymmetry of the distribution. The kurtosis does not disappear as *n* grows indicating the presence of long-tails. Note that the kurtosis also reaches an apparent minimum at *n* = 7.

shown). This is expected since the rise time of a single gene is almost independent of the preceding events unlike the period, which is an aggregated quantity and therefore more susceptible to propagated noise. The investigation of the noise characteristics of networks of transcriptional oscillators will be the object of further study.

Discussion

The present work presents a detailed implementation of the DCFTP-SSA that could be integrated with other packages in Computational Systems Biology. We have also provided a mathematical proof of the algorithm with an explicit statement of the limits of its applicability. This detailed description is key to the extension of the algorithm to a wider class of systems. Specifically, the DCFTP-SSA can be applied to conversion reactions of the type $A \to B$ with the realistic assumption that the monotone propensity function only depends on n_A. Unfortunately, the extension to encompass bimolecular reactions of the type $A + B \to C$ does not seem to be trivial, since the partial ordering used in this paper will not be preserved and there is no dominating process with known stationary distribution readily available. The latter problem can be addressed partially by using the CFTP under the approximation that there is an upper bound on the number of molecules in the state space. If the bound is chosen to be large enough, it can be shown numerically that the error will be negligible. However, this approximate method will not carry the guarantees of stationarity that the DCFTP-SSA provides.

From the numerical viewpoint, the DCFTP-SSA is guaranteed to converge almost surely in finite time, but there is no upper bound on the coalescence times T_c. Our numerics show that the distribution of coalescence times can be long-tailed when the structure of the stationary distribution is complex (Fig. 3b).

If a simulation is interrupted prematurely by an impatient user, the final sample will be biased. An alternative perfect sampling scheme is the FMMR algorithm [14], which uses rejection sampling to circumvent this problem. Our experience has shown that typically a small fraction of runs takes a very long time to converge. Being able to remove these would speed up the algorithm significantly. The bimodal example illustrates this point: if we were able to place a cut-off after the first mode, a large portion simulations would be accepted and at the same time there would be a significant save in terms of both CPU time and memory. As indicated by the examples in this paper, it is important to note that the DCFTP-SSA does not present obvious problems with scalability, as the overhead incurred to provide a certificate of stationarity is moderate. Although the computational cost of the algorithm depends on the intrinsic structure of the network, we have applied the DCFTP-SSA to various networks with several tens of variables.

In addition to producing guaranteed sampling from the stationary distribution, the DCFTP-SSA can be used to provide initial conditions for ordinary SSA runs. Since any Markov process started from stationarity will remain there for all future times, these runs are guaranteed to represent the stationary time-traces of the system. This is important for the numerical characterization of properties such as escape times and autocorrelation times of systems with high variability, e.g., with underlying multi-stable, oscillatory or excitatory behaviour [15].

Conclusion

The SSA is an exact procedure to sample the time-dependent probability distribution of the ME of general chemical reaction networks at all times [7,8]. However, it provides no guarantees when the aim is sampling from the stationary distribution. The DCFTP-SSA presented here addresses this problem for a class of networks of relevance to genetic and enzymatic regulation. Our algorithm provides guaranteed stationary sampling and thus removes one of the sources of uncertainty in stochastic simulations. This can aid in the characterization of regulatory circuits and in the testing of model hypotheses for these systems.

Authors' contributions

MH and MB developed the method, completed the proof and wrote the paper. MH implemented the algorithm for the simulations.

Acknowledgements

We would like to thank Matias Rantalainen, Vincent Rouilly and Sophia Yaliraki for valuable discussions and feedback. Research funded by the UK EPSRC through the Life Science Interface and the Mathematics Panels.

References

1. Elowitz MB, Levine AJ, Siggia ED, Swain PS: **Stochastic gene expression in a single cell.** *Science* 2002, **297:**1183-1186.
2. Raser JM, O'Shea EJ: **Control of stochasticity in eukaryotic gene expression.** *Science* 2004, **304(5678):**1811-1814.
3. Raj A, Peskin CS, Tranchina D, Vargas DY, Tyagi S: **Stochastic mRNA synthesis in mammalian cells.** *PLoS Biol* 2006, **4(10):**e309.
4. van Kampen NG: *Stochastic processes in physics and chemistry* 2nd edition. Elsevier; 1992.
5. McAdams HH, Arkin A: **Stochastic mechanisms in gene expression.** *Proc Natl Acad Sci U S A* 1997, **94(3):**814-819.
6. Paulsson J: **Summing up the noise in gene networks.** *Nature* 2004, **427:**415-419.
7. Gillespie DT: **A general method for numerically simulating the stochastic time evolution of coupled chemical reactions.** *Journal of Computational Physics* 1976, **22:**403-434.
8. Gillespie DT: **A rigorous derivation of the chemical master equation.** *Physica A* 1992, **188:**404-425.
9. Norris JR: *Markov chains* Cambridge University Press; 1999.
10. Valeriani C, Allen RJ, Morelli MJ, Frenkel D, ten Wolde PR: **Computing stationary distributions in equilibrium and nonequilibrium systems with forward flux sampling.** *Journal of Chemical Physics* 2007, **127:**.

11. Propp JG, Wilson DB: **Exact sampling with coupled Markov chains and applications to statistical mechanics.** *Random structures and algorithms* 1996, **9:**223-252.
12. Thönnes E: **A primer on perfect simulation.** *Springer Lecture Notes in Physics, Springer* 2000, **554:**349-378.
13. Kendall WS, Møller J: **Perfect simulation using dominating processes on ordered spaces with application to locally stable point processes.** *Advances in Applied Probability* 2000, **32:**844-865.
14. Fill JA, Machida M, Murdoch DJ, Rosenthal JS: **Extension of Fill's perfect rejection sampling algorithm to general chains.** *Random Structures and Algorithms* 2000, **9:**223-252.
15. Hemberg M, Barahona M: **Perfect Sampling of the Master Equation with Applications to Gene Regulatory Networks.** *Biophysical Journal* 2007, **93(2):**.
16. Lindvall T: *Lectures on the coupling method* John Wiley & Sons, Inc; 1992.
17. Doob JL: **Markoff chains – Denumerable case.** *Transactions of the American Mathematical Society* 1945, **58(3):**455-473.
18. Rao CV, Arkin AP: **Stochastic chemical kinetics and the quasi steady-state assumption: Applications to the Gillespie algorithm.** *Journal of Chemical Physics* 2003, **118(11):**4999-5011.
19. Gadgil C, Lee CH, Othmer HG: **A stochastic analysis of first-order reaction networks.** *Bulletin of Mathematical Biology* 2005, **67(5):**901-946.
20. Häggström O, Nelander K: **Exact sampling from anti-monotone systems.** *Statistica Neerlandica* 1998, **52(3):**360-380.
21. Widder S, Schicho J, Schuster P: **Dynamic patterns of gene regulation I: simple two gene systems.** *Journal of Theoretical Biology* 2007, **246(3):**395-419.
22. Gardner TS, Cantor CR, Collins JJ: **Construction of a genetic toggle switch in *Escherichia coli*.** *Nature* 2000, **403:**339-343.
23. Elowitz MB, Leibler S: **A synthetic oscillatory network of transcriptional regulators.** *Nature* 2000, **403:**335-339.
24. Goldbeter A: **Computational approaches to cellular rhythms.** *Nature* 2002, **420:**238-245.
25. Muller S, Hofbauer J, Endler L, Flamm C, Widder S, Schuster P: **A generalized model of the repressilator.** *Journal of Mathematical Biology* 2006, **53:**905-937.
26. Mitzenmacher M, Upfal E: *Probability and computing: randomized algorithms and probabilistic analysis* Cambridge University Press; 2005.