Efficient stochastic simulation of chemical kinetics networks using a weighted ensemble of trajectories

Rory M. Donovan, Andrew J. Sedgewick, James R. Faeder,[a] and Daniel M. Zuckerman[b]
Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, Pennsylvania 15260, USA

(Received 21 March 2013; accepted 29 August 2013; published online 20 September 2013)

We apply the "weighted ensemble" (WE) simulation strategy, previously employed in the context of molecular dynamics simulations, to a series of systems-biology models that range in complexity from a one-dimensional system to a system with 354 species and 3680 reactions. WE is relatively easy to implement, does not require extensive hand-tuning of parameters, does not depend on the details of the simulation algorithm, and can facilitate the simulation of extremely rare events. For the coupled stochastic reaction systems we study, WE is able to produce accurate and efficient approximations of the joint probability distribution for all chemical species for all time *t*. WE is also able to efficiently extract mean first passage times for the systems, via the construction of a steady-state condition with feedback. In all cases studied here, WE results agree with independent "brute-force" calculations, but significantly enhance the precision with which rare or slow processes can be characterized. Speedups over "brute-force" in sampling rare events via the Gillespie direct Stochastic Simulation Algorithm range from $\sim 10^{12}$ to $\sim 10^{18}$ for characterizing rare states in a distribution, and $\sim 10^2$ to $\sim 10^4$ for finding mean first passage times. © *2013 AIP Publishing LLC*. [http://dx.doi.org/10.1063/1.4821167]

I. INTRODUCTION

Stochastic behavior is an essential facet of biology, for instance in gene expression, protein expression, and epigenetic processes.[1–14] Stochastic chemical kinetics simulations are often used to study systems biology models of such processes.[15–17] One of the more common stochastic approaches, the one employed in the present study, is the stochastic simulation algorithm (SSA), also known as the Gillespie algorithm.[15,18,19]

As stochastic systems biology models approach the true complexity of the systems being modeled, it quickly becomes intractable to investigate rare behaviors using naïve ("brute-force") simulation approaches. By their very nature, rare events occur infrequently; confoundingly, rare events are often those of most interest. For example, the switching of a bistable system from one state to another may happen so infrequently that running a stochastic simulation long enough to see transitions would be (extremely) computationally prohibitive.[20] This impediment only grows as model complexity increases, and as such it poses a serious hurdle for systems models as they grow more intricate.

Several approaches to speeding up the simulation of rare events in stochastic chemical kinetic systems exist. A variety of "leaping" methods can, by taking advantage of approximate time-scale separation, accelerate the SSA itself.[21–28] Kuwahara and Mura's weighted stochastic simulation (wSSA) method[29] was refined by Gillespie and co-workers,[30–33] and is based on importance sampling. The forward flux sampling method of ten Wolde and co-workers[20,34–38] uses a series of interfaces in state-space to reduce computational effort, as does the non-equilibrium umbrella sampling approach.[39,40]

Rare event sampling is also an active topic in the field of molecular dynamics simulations, and many approaches have been proposed. Of the approaches that do not irreversibly modify the free energy landscape of the system, some notable methods include dynamic importance sampling,[41] milestoning,[42] transition path sampling,[43] transition interface sampling,[44] forward flux sampling,[36] non-equilibrium umbrella sampling,[40] and weighted ensemble sampling.[45–52] For a summary of these methods, see Ref. 53. Many of the ideas behind these techniques are not exclusive to molecular dynamics simulations, and can be adapted to studying stochastic chemical kinetic models. For example, dynamic importance sampling seems to be closely related to wSSA.

Because of its relative simplicity and potential efficiency in sampling rare events, we apply one of these methods, the weighted ensemble algorithm (WE) to well-established model systems of stochastic kinetic chemical reactions. These models range in complexity from one species and two reactions to 354 species and 3680 reactions. For the systems studied, WE proves many orders of magnitude faster than SSA simulation alone, offers linear parallel scaling, returns full distributions of desired species at arbitrary times, and can yield mean first passage times (MFPTs) via the setup of a feedback steady-state.

II. METHODOLOGY

The methods employed are described immediately below, while the models are specified in Sec. III.

[a] faeder@pitt.edu. URL: http://www.csb.pitt.edu/Faculty/Faeder/.
[b] ddmmzz@pitt.edu. URL: http://www.csb.pitt.edu/Faculty/zuckerman.

A. Stochastic chemical kinetics and BioNetGen

Stochastic chemical kinetics occupies a middle-ground in the realm of chemical simulation, between very explicit, and costly, molecular dynamics (MD) simulations and the deterministic formalism of reaction rate equations (RREs). Stochastic chemical kinetics attempts to account for the randomness inherent in chemical reactions, without trying to explicitly model the spatial structure of the reacting species. It is many orders of magnitude faster than MD simulations, but much slower than the RRE approach. Stochastic chemical kinetics is ideal to use for modeling the effects of low concentrations (or copy numbers) of chemical reactants, while ignoring the effects of specific spatial distribution.

Stochastic chemical kinetics models can be solved exactly for sufficiently simple systems using the Chemical Master Equation (CME), and approximately (for all systems) using Gillespie's direct stochastic simulation algorithm (SSA).[15,18,19] The SSA samples the CME exact solution by modeling stochastic chemical kinetics in a straightforward manner, and yields trajectories of species concentrations that converge to the RRE method in the limit of large amounts of reactants. In brief, the SSA iteratively and stochastically determines which reaction fires at what time by sampling from the exponential distribution of waiting times between reactions. For a detailed explanation of the SSA, see Ref. 15.

We employ the rule-based modeling and simulation package BioNetGen[54] to simulate both our toy and complex models. Rule-based modeling languages allow the specification of biochemical networks based on molecular interactions. Rules that describe those interactions can be used to generate a reaction network that can be simulated either as RREs or using the SSA, or the rules can be used directly to drive stochastic chemical kinetics simulations. BioNetGen has been applied to a variety of systems, such as the aggregation of membrane proteins by cytosolic cross-linkers in the LAT-Grb2-SOS1 system,[55] the single-cell quantification of IL-2 response by effector and regulatory T cells,[56] the analysis and verification of the HMGB1 signaling pathway,[57] the role of scaffold number in yeast signaling systems,[58] and the analysis of the roles of Lyn and Fyn in early events in B cell antigen receptor signaling.[59] We employ BioNetGen's implementation of the direct SSA to propagate the dynamics in our systems.

B. Weighted ensemble

WE is a general-purpose protocol used in molecular dynamics simulations[46–48,50–52] that we adapt here to the efficient sampling of dynamics generated by chemical kinetic models. In brief, WE employs a strategy of "statistical natural selection" using quasi-independent parallel simulations which are coupled by the intermittent exchange of information. The intermittency leads directly to linear parallel scaling. Importantly, the simulations are coupled via configuration space (essentially the "phase space" of the system in physics language or the "state-space" in cell and population modeling). This type of coupling permits both efficiency and a large degree of scale independence. The efficiency results from distributing trajectories to typically undersampled parts of the space, while scale independence is afforded because every type of system has a configuration or state-space.

WE's strategy of statistical natural selection or statistical ratcheting is schematized in Fig. 1. First, the state-space is divided/classified into non-overlapping "bins" which are

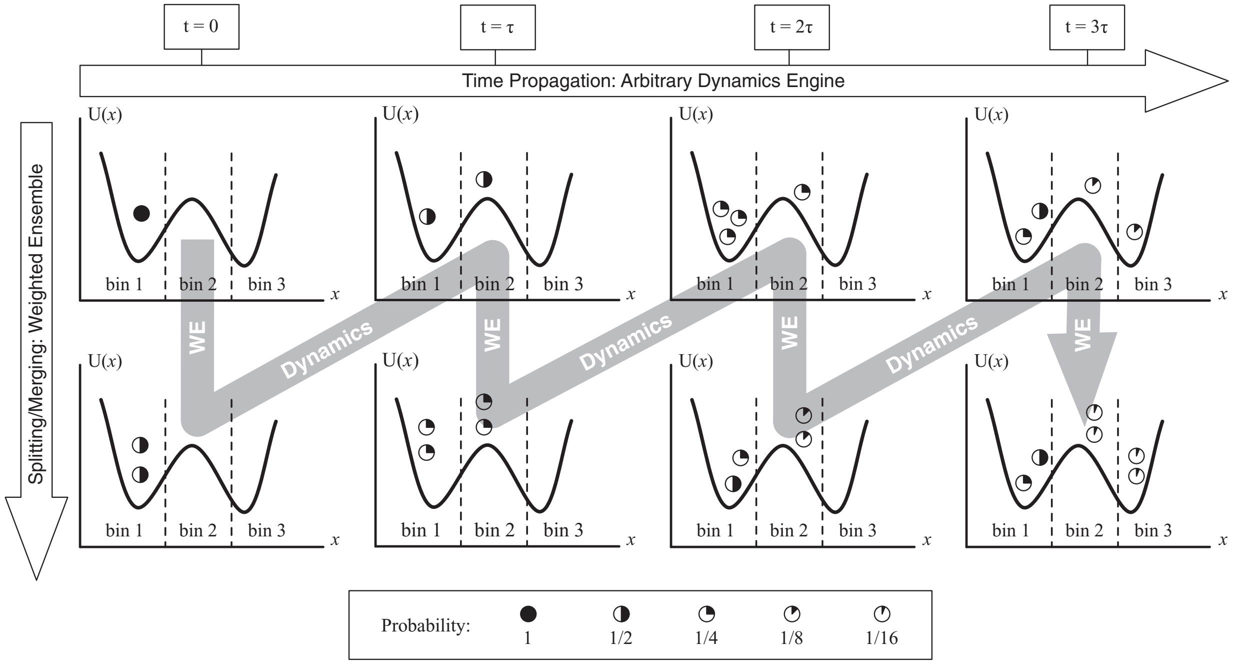
t = 0
t = τ
t = 2τ
t = 3τ
Time Propagation: Arbitrary Dynamics Engine
Splitting/Merging: Weighted Ensemble
U(x)
bin 1
bin 2
bin 3
x
WE
Dynamics
Probability:
1
1/2
1/4
1/8
1/16

FIG. 1. Weighted ensemble (WE) simulation depicted for a configuration/state-space divided into bins. Multiple trajectories are run using any dynamics software (here we use the SSA in BioNetGen) and checked every τ for bin location. Trajectories are assigned weights (symbols — see legend) that sum to one and are split and combined according to statistical rules that preserve unbiased kinetic behavior.

typically static, although dynamic and adaptive tessellations are possible.[47] A target number of trajectories, M_{targ}, is set for each bin. Multiple trajectories are initiated from the desired initial condition and each is assigned a weight (probability) so that the sum of weights is one. Trajectories are then simulated independently according to the desired dynamics (e.g., molecular dynamics or SSA) and checked intermittently (every τ units of time) for their location. If a trajectory is found to occupy a previously unoccupied bin, that trajectory is split and replicated to obtain the target number of copies, M_{targ}, for the bin. The sum of these newly spawned trajectories' weights must sum to the weight of the parent trajectory, so in a splitting event, if the original parent trajectory has weight w, the weight of each daughter is set to w/M_{targ}. If a bin is occupied by more than the target number, trajectories must be pruned in a statistical fashion maintaining the sum of weights. Specifically, the two lowest weight trajectories are "merged" by randomly selecting one of them to survive, with probability proportional to their weights, and the surviving trajectory absorbs the weight of the pruned one. This resampling process is repeated as needed, and maintains an exact statistical representation of the evolving distribution of trajectories.[47]

Setting up a WE simulation requires selection of state-space binning, trajectory multiplicity, and timing parameters. In our simulations, we chose to divide the state-space of an N-dimensional system into one- or two-dimensional regular grids of non-overlapping bins. It is possible to use non-Cartesian bins, and to adaptively change the bins during simulation,[47,50] but for simplicity we did not pursue any such optimization. Specific parameter choices for each model are given in Sec. III.

The weighted ensemble can be outlined fairly concisely. Let M_{targ} be the target number of segments in each bin, N_{bins} the number of bins, whose geometry are defined by the grid G_{grid}, τ the time-step of an iteration of WE, and N_{iters} the total number of iterations of WE. The WE procedure also requires an initial state of the system, $\mathbf{x}_0$, which in our case is a list of the concentrations of all the chemical species in the system. Given these parameters, the WE algorithm is then:

```
procedure WE($N_{iters}$, $\tau$, $G_{grid}$, $M_{targ}$, $\mathbf{x}_0$)
  for $i = 1 \ldots N_{iters}$ do
    for each populated bin in $G_{grid}$ do
      propagate dynamics for all trajectories
      update bin populations
    for each bin in $G_{grid}$ do
      if bin population $= 0$ or $M_{targ}$ then
        do nothing
      else if bin population $< M_{targ}$ then
        replicate trajectories until bin pop. $= M_{targ}$
        maintain sum of weights in each bin
      else if bin population $> M_{targ}$ then
        merge trajectories until bin pop. $= M_{targ}$
        maintain sum of weights in each bin
    save coordinates and weights of each trajectory
  return trajectory coordinates & weights for each iter.
```

The replicating and merging of trajectories in the above algorithm are done randomly, according to the weight of each trajectory segment in a given bin, which has been shown not to bias the dynamics of the ensemble.[45,48]

When WE is used to manage an ensemble of trajectories, there are two time-scales of immediate concern: the period at which trajectory coordinates are saved, and the period τ at which ensemble operations are performed. These two time-scales can be different, but for simplicity we set them to be the same, and select τ such that it is greater than the inverse of the average event firing rate for the SSA. When we refer to the time-step, or iteration of a process, we are referring to the τ of Fig. 1.

WE can be employed in a variety of modes to address different questions. Originally developed to monitor the time evolution of arbitrary initial probability distributions,[45] i.e., non-stationary non-equilibrium systems, WE was generalized to efficiently simulate both equilibrium and non-equilibrium steady-states.[48] In steady-state mode, mean first passage times (MFPTs) can be estimated rapidly based on simulations much shorter than the MFPT using a simple rigorous relation between the flux and MFPT.[48] Steady-states can be attained rapidly, avoiding long relaxation times, by using the inter-bin rates computed during a simulation to estimate bin probabilities appropriate to the desired steady-state; trajectories are then reweighted to conform to the steady-state bin probabilities.[48] Both of these methods are described in more detail below.

1. Basic WE: Probability distribution evolving in time

Perhaps the simplest use of a weighted ensemble of trajectories is to better sample rare states as a system evolves in time, specifically the states corresponding to extreme values of the binning coordinate. The SSA itself samples the exact distribution, but its sampling is concentrated about the mode(s) of the distribution. The SSA naturally — and correctly — samples rare states infrequently. By using WE to split up the state-space, however, one can resample the distribution at every time step τ, selecting those trajectories that advance along a progress coordinate for more detailed study, but doing so without applying any forces or biasing the trajectories or the distribution. Essentially, WE appropriates much of the effort that brute-force SSA devotes to sampling the central component of the distribution, repurposing it to obtain better estimates of the tails.

This basic use of WE requires none of the "tricks" we apply in later sections, such as using reweighting techniques to accelerate obtaining a steady-state. We apply basic WE to some of our systems — particularly, but not exclusively, to those that are not bistable.

2. Steady-state

The mean first passage time (MFPT) from state A to state B is a key observable. It is equal to the inverse of the flux (of probability density) from state A to state B in steady-state,[60]

$$\text{MFPT}_{A\to B} = \frac{1}{\text{Flux}_{ss}(A \to B)}. \tag{1}$$

This relation provides the weighted ensemble approach the ability to calculate MFPTs in a straightforward manner. During a WE run, when any trajectories (and their associated weights) reach a designated target area of state-space (or "state *B*"), they are removed and placed back in the initial state ("state *A*"). Eventually, such a process will result in a steady-state flow of probability from state *A* to state *B* that does not change in time (other than with stochastic noise).

Reweighting. The waiting time to obtain a steady-state constrains the efficiency of obtaining a MFPT by measuring fluxes via equation (1). This waiting time can vary from the relatively short time scale of intra-state equilibration for simple systems, to much longer time-scales, on the order of the MFPT itself for more complicated systems. To reduce this waiting time, we use the steady-state reweighting procedure of Bhatt *et al.*[48] This method measures the fluxes between bins to obtain a rate-matrix for transitions between bins, and uses a Markov formulation to infer a steady-state distribution from the (noisy) data available.

For instance, let $\{w_i\}$ be the set of bin weights (i.e., the sum of the weights of the trajectories in each bin), and let $\{w_i^{ss}\}$ be the set of steady-state values of the bin weights. If f_{ij} is the flux of weight into bin *i* from bin *j*, then in steady-state, the flux out of a bin is equal to the flux into it,

$$\frac{\mathrm{d}w_i^{ss}}{\mathrm{d}t} = \sum_j \left(f_{ij}^{ss} - f_{ji}^{ss}\right) = \sum_j \left(k_{ij}w_j^{ss} - k_{ji}w_i^{ss}\right) = 0. \tag{2}$$

Since the flux of weight into bin *i* from bin *j* is the product of a (constant) rate and the (current) weight in a bin, i.e., $f_{ij} = k_{ij}w_j$ (true for both steady state and not), we can use Eq. (2) to find the inter-bin rates. By measuring the inter-bin fluxes and the bin weights, we can approximately infer the transition rates, and then find a set of weights that satisfy Eq. (2). Once the set of bin weights is found, the weights of the individual trajectories in the bins are rescaled commensurately. This reweighting process should not be confused with a resampling process (such as basic WE splitting and merging) which does not change the distribution.

The steady-state distribution of weights thus inferred is not necessarily the true steady-state of the system, but tends to be closer to it than the distribution was prior to reweighting, and an iterative application of this procedure can converge to the true distribution fairly rapidly. In practice, it has been shown to accelerate the system's evolution to a true steady-state by orders of magnitude in some cases.[48]

C. Estimation of computational efficiency

Since it is important to assess new approaches quantitatively, we compare the speedup in computing time from weighted ensemble to a brute-force simulation (i.e., SSA). For a given observable (e.g., the fraction of probability in a specified tail of the distribution) and a desired precision, we estimate the efficiency using the ratio:

$$E := \frac{\text{dynamics time in brute-force SSA}}{\text{dynamics time in WE-SSA}}. \tag{3}$$

Since both WE and brute-force use the same dynamics engine/software, we can estimate the speedup of WE over brute-force by just keeping track of how much total "dynamics time" was simulated in each. We employ this measure when estimating the advantage of using WE to investigate the tails of probability distributions, as well as for finding MFPTs in bistable systems.

Another measure of efficiency we employ for MFPT estimation gauges how fast WE attains a result that is within 50% of the true result (determined from exact or extensive brute-force calculation):

$$E_{50\%} := \frac{\text{dynamics time in brute-force SSA to get} \pm 50\%\ \text{exact}}{\text{dynamics time in WE-SSA to get} \pm 50\%\ \text{exact}}. \tag{4}$$

This is an assessment of how well WE can extract rough estimates of long time-scale behavior from simulations that are much shorter than those timescales.

Brute-force SSA simulations can be run for long times without seeing a transition from one macro-state to another. To take account of the brute-force simulations where no transitions occurred we use a maximum likelihood estimator for the transition time, based on an exponential distribution of waiting times,[61] which is a valid approximation for the one-dimensional and two-state systems studied below:

$$\begin{aligned} \mu_{MLE} &= \left(1 - \frac{n}{N}\right)T + \frac{1}{n}\sum_{i=1}^{n} t_i, \\ \sigma_\mu &= \frac{\mu_{MLE}}{\sqrt{n}}, \end{aligned} \tag{5}$$

where *T* is the length of the brute-force simulations, *N* is the number of these simulations performed, *n* is the number of these simulations in which a transition from one state to another is observed, and t_i are the times at which the transition is observed.

D. Limitations of our implementation

We used two different implementations of the weighted ensemble framework: WESTPA, written in Python, is the most feature-rich and stable,[52] and is available at http://chong.chem.pitt.edu/WESTPA. Another, written by Zhang *et al.*[46] and modified by us, is written in C, and is faster though less robust; it is available at http://donovanr.github.com/WE_git_code.

Weighted ensemble (WE), as a scripting-level approach, inherently adds some overhead to the runtime of the dynamics. This overhead, in theory, is quite minimal: stopping, starting, merging, and splitting trajectories are not computationally costly operations. A key issue in practical implementations, though, is how long the algorithm actually takes to run, i.e., the wall-clock running-time for dynamics (here, the SSA).

In practice, overhead can be significant for very simple systems, for the sole reason that reading and writing to disk takes so much time compared to how long it takes to run the dynamics of small models. In our implementation, data are passed from the dynamics engine to WE by reading and writing files to disk. This handicap is an artifact of our interface, which could, with minimal work, be modified to something more efficient. As a proof-of-principle, the version of WE written in C was modified, for the Schlögl reactions and the futile cycle, to contain hard-coded versions of the Gillespie direct algorithm for those systems, so as to obviate the I/O between WE and BNG. With these modifications, it was difficult to ascertain any significant overhead costs at all, and our runs completed in a matter of seconds. We also note that as model complexity increases and more time is devoted to dynamics, the overhead problem becomes negligible. Practical applications of WE will, by nature, target models where dynamics are expensive, rather than toy models, where they are cheap.

III. MODELS & RESULTS

We study four different models, ranging in complexity from two chemical reactions governing one chemical species to 3680 reactions governing 354 species. The models employ coupled stochastic chemical reactions, which we implement and simulate in BioNetGen using the SSA.[15,18,19] As depicted in Fig. 1, these simulations are, in turn, managed by a weighted ensemble procedure.

A. Enzymatic futile cycle

1. Model

The enzymatic futile cycle is a simple and robust model that can, in certain parameter regimes, exhibit qualitatively different behavior due to stochastic noise.[62,63] This signaling motif can be seen in biological systems including GTPase cycles, MAPK cascades, and glucose mobilization.[62,64,65]

The enzymatic futile cycle studied here is modeled by

$$\begin{aligned} E_1 + S_1 &\underset{k_f}{\overset{k_f}{\rightleftharpoons}} B_1 \xrightarrow{k_s} E_1 + S_2, \\ E_2 + S_2 &\underset{k_f}{\overset{k_f}{\rightleftharpoons}} B_2 \xrightarrow{k_s} E_2 + S_1, \end{aligned} \tag{6}$$

where $k_f = 1.0$ and $k_s = 0.1$. Here S_1 can bind to its enzyme E_1, and in the bound form, B_1, (i.e., $B_1 = E_1 \cdot S_1$), it can be converted to S_2, and then dissociate (and similarly for $S_2 \rightarrow S_1$). The total amount of substrate, $S_1 + S_2$, is conserved, as are the amounts of the different enzymes E_1 and E_2, of which is supplied only one of each kind, thus $(S_1 + B_1) + (S_2 + B_2) = 100$. Following Kuwahara and Mura,[29] in the specific system we look at, we set $S_1 + S_2 = 100$ and $E_1 + B_1 = E_2 + B_2 = 1$.

Thus constrained, the above system of reactions can be solved by an approximately 400-state chemical master equation (CME), to obtain an exact probability density for all times when initialized from an arbitrary starting point. We start the system at $S_1 = S_2 = 50$ and $E_1 = E_2 = 1$, and are interested in the probability distribution of S_1 after 100 s, that is, $P(S_1 = x, t = 100)$.

2. WE parameters

The WE data were generated using 101 bins of unit width on the coordinate S_1. We employed 100 trajectory segments per bin that were run for 100 iterations of a $\tau = 1$ s time-step, with no reweighting events. The brute-force data are from 10,200 100-s runs, which is an equivalent amount of dynamics to compute as the single WE run, if all the bins were full all the time. However, since the bins take some time to fill up, the WE run employed only 840,000 1-s segments, which makes the comparison to brute-force SSA more than fair.

3. Results

Figure 2 shows that the brute-force SSA is unable to sample values of S_1 much outside the range $30 < S_1 < 70$, whereas the WE method is able to accurately sample the entire distribution. Waiting for the brute-force approach to sample the tails would take $\sim 1/P(\text{tail}) \sim 1/10^{-23} \sim 10^{23}$ brute-force runs. With a conservative estimate of $\sim 10^4$ runs per second, it would take $\sim 10^{19}$ s, or many times the age of the universe, for brute-force SSA to sample the tails at all. WE takes 2–3 s to sample them (note the comparison to exact distribution provided by the CME), for an approximate efficiency increase $E \sim 10^{18}$.

For the sake of clarity, error-bars were omitted from Fig. 2. Over most of the data range, the error is too small to see on the plot. In the tails (of both SSA and WE-SSA), the error is not computable from a single run, since there are plot points comprised of only a single trajectory. The error in the

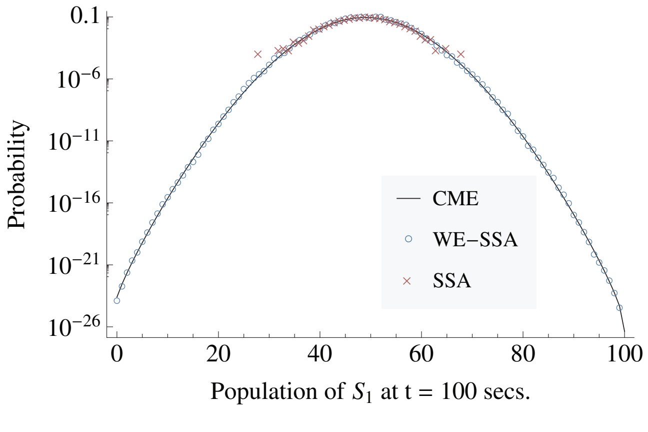

FIG. 2. The probability distribution of S_1 in the enzymatic futile cycle system, after $t = 100$ s, when initialized from a delta function at $S_1 = 50$, $E_1 = E_2 = 1$ at $t = 0$. The exact solution, procured via the chemical master equation (CME), is compared to data obtained using the SSA in a weighted ensemble run (WE-SSA), and to ordinary SSA, when each are given equal computation time. WE data are from a single run. Error bars are not plotted; for a discussion of uncertainties, see Sec. III A 3. The noise in the tail of the SSA points and the gap in coverage are indicative of the high variance of SSA for rare events.

estimate of the distribution can be inferred visually from the data's departure form the CME exact solution. For SSA, however, generating uncertainties far all values is essentially impossible. When computing quantitative observables reported below, we employ multiple independent runs to procure standard errors in our estimate.

From the distribution, we are able to read off useful statistics. For instance, in the spirit of Kuwahara and Mura,[29] and Gillespie *et al.*,[30] one might desire to know the probability of the futile cycle to have a value of $S_1 > 90$ at $\Delta t = 100$, which we will denote as $p_{>90}$. Since WE gives an accurate estimate of $P(x, t)$ on an arbitrarily precise spatiotemporal grid, all that is required to find $p_{>90}$ is to sum up the area under the state of interest: we find $p_{>90} = 2.47 \times 10^{-18} \pm 3.4 \times 10^{-19}$ at one standard error, as computed from ten replicates of the single WE run plotted in Fig. 2. The CME gives an exact value of 2.72×10^{-18}. Following Gillespie *et al.*,[30] the approximate number of normal SSA runs needed to estimate this observable with comparable error is $n^{\mathrm{SSA}}_{p_{>90}} = p_{>90}/(\sigma_{p_{>90}})^2 = 2.72 \times 10^{-18}/(3.4 \times 10^{-19})^2 = 2.4 \times 10^{19}$ SSA runs. Using ten replicate runs, WE is able to sample it using a total of 8,317,000 trajectory segments, which is computationally equivalent to 83,170 brute-force trajectories, resulting in an increase in sampling efficiency by a factor of $E \sim 2.4 \times 10^{19}/83{,}170 \sim 3 \times 10^{14}$ for this observable at this level of accuracy. Since WE gives the full distribution, from the same data we can also find other rare event statistics. For example, we might also wish to compute the probability that $S_1 \leq 25$ at $\Delta t = 100$, which we will call $p_{\leq 25}$. From the same data described in the preceding paragraph, we can find $p_{\leq 25} = 1.35 \times 10^{-7} \pm 1.5 \times 10^{-8}$ at one standard error. The CME gives an exact value of 1.26×10^{-7}. Again, estimating the number of SSA runs necessary to determine the same observable with similar accuracy as in Ref. 30, we find $n^{\mathrm{SSA}}_{p_{\leq 25}} = p_{\leq 25}/(\sigma_{p_{\leq 25}})^2 = 1.26 \times 10^{-7}/(1.5 \times 10^{-8})^2 = 6.0 \times 10^{8}$ SSA runs. Our weighted ensemble calculation used the computational equivalent of 83,170 brute-force trajectories, resulting in an increase in sampling efficiency by a factor of $E \sim 6.0 \times 10^{8}/83{,}170 \sim 7 \times 10^{3}$ for this observable at this level of accuracy.

Kuwahara and Mura[29] and Gillespie *et al.*[30] define a quantity closely related to those computed above: the probability of a system to pass from one state to another in a certain time: $P(x_i \to x_f|\Delta t)$.[29–33] This is subtly different than just measuring areas under a distribution, since a trajectory is terminated, and removed from the ensemble, if it successfully reaches the target. To make a direct comparison to this statistic, we implemented in WE an absorbing boundary condition at the target state $S_1 = 25$. Using ten WE runs, we find that the probability of reaching the $S_1 = 25$ within 100 s, which we call $p_{\mathrm{abs}(25)}$, is $1.61 \times 10^{-7} \pm 1.93 \times 10^{-8}$ at one standard error. The CME with an absorbing boundary condition at $S_1 = 25$ gives an exact value of 1.738×10^{-7}. As above, we estimate the number of brute-force SSA runs needed to attain a comparable estimate as $n^{\mathrm{SSA}}_{p_{\mathrm{abs}(25)}} = p_{\mathrm{abs}(25)}/(\sigma_{p_{\mathrm{abs}(25)}})^2 = 1.738 \times 10^{-7}/(1.93 \times 10^{-8})^2 = 4.66 \times 10^{8}$ SSA runs. In total, the ten WE runs used 6,342,600 trajectory segments, which is computationally equivalent to 63,426 brute-force trajectories. This yields an increase in efficiency of $E \sim 4.66 \times 10^{8}/63{,}426 \sim 7 \times 10^{3}$ for this observable at this level of accuracy.

For the above statistic, Gillespie *et al.* report an efficiency gain of 7.76×10^{5} over brute-force SSA.[30] Direct comparisons of efficiency gain between wSSA and WE are difficult, even when focussing on the same observable in the same system. Many factors contribute to this, notably that WE is not optimized in a target-specific manner, as wSSA historically has been. Further, in its simplest form, WE yields a full space/time distribution of trajectories, from which it is possible to calculate rare event statistics for arbitrary states, which need not be specified in advance.

B. Schlögl reactions

1. Model

The Schlögl reactions are a classic toy-model for benchmarking stochastic simulations of bistable systems.[66–68] They are two coupled reactions with one dynamic species, X:

$$\begin{aligned} A + 2X &\underset{k_2}{\overset{k_1}{\rightleftharpoons}} 3X, \\ B &\underset{k_4}{\overset{k_3}{\rightleftharpoons}} X, \end{aligned} \tag{7}$$

where $k_1 = 3 \times 10^{-7}$, $k_2 = 10^{-4}$, $k_3 = 10^{-3}$, $k_4 = 3.5$, $A = 10^{5}$, and $B = 2 \times 10^{5}$. The species A and B are assumed to be in abundance, and are held constant. Both the mean first passage times and the time-evolution of arbitrary probability distributions can be computed exactly.[67]

2. WE parameters

The WE data in Fig. 3 were generated using 802 bins of unit width, 100 trajectory segments per bin, a time-step $\tau = 0.05$ s, and run for 100 iterations of that time-step, with no

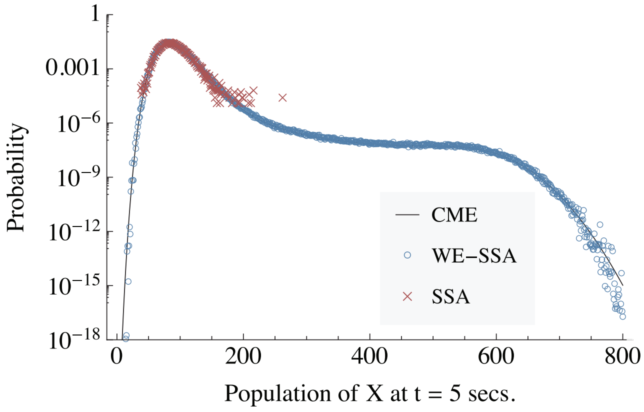

FIG. 3. The probability distribution of X in the Schlögl system, at $t = 5$ s, when initialized from a delta function at $X = 82$. The exact solution from the chemical master equation (CME) is compared to data obtained using the SSA in a weighted ensemble run (WE-SSA), and to ordinary SSA, when each are given equal computational time. WE data are from a single run. For a discussion of uncertainties, see Sec. III B 3. The noise in the tail of the SSA points and the gap in coverage are indicative of the high variance of SSA for rare events.

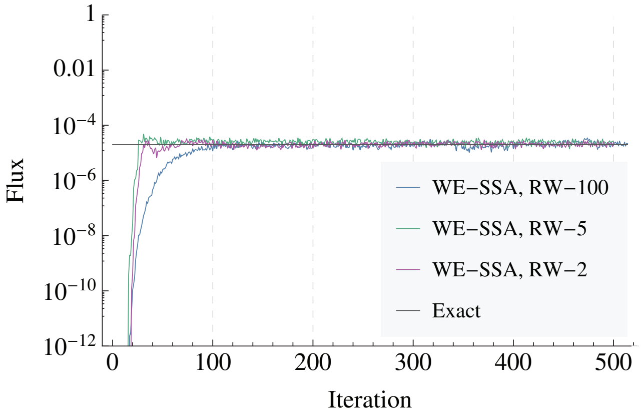

FIG. 4. The flux of probability into the target state ($X \geq 563$) for the Schlögl system. The exact value is compared to WE results, for reweighting (RW) periods of every 100, 5, and 2 iterations. The inverse of the flux gives the mean first passage time by Eq. (1).

reweighting events. The brute-force data are from 80,200 5-s runs, which is an equivalent amount of dynamics as a single WE run, if all bins are always full. Were that the case, the WE run would compute dynamics for 8,020,000 trajectory segments; in our case, the WE simulation ran 7,047,300 trajectory segments, which makes the comparison to brute-force more than fair.

The WE data in Fig. 4 were generated using 80 bins of width 10, with 32 trajectory segments per bin, a time-step $\tau = 0.1$ s, run for 500 iterations of that time-step. Reweighting events (see Sec. II B 2) were applied every 100, 5, and 2 iterations for the data labeled “RW-100”, “RW-5”, and “RW-2”, respectively.

3. Results

Figure 3 shows how the results of both a brute-force (BF) approach, and the WE approach compare to the exact solution,[67] when each employs the same amount of dynamics time. We start the Schlögl system with $X = 82$, i.e., the PDF is initially a delta function at $X = 82$. To investigate rare transitions, we study the PDF at time $t = 5$ s. WE is able to accurately sample almost the entire distributions, even over the potential barrier near $X = 250$, while the BF approach is limited to sampling only high probability states. The Schlögl system is bistable, with states centered at $X = 82$ and $X = 563$, and a potential barrier between them, peaked at $X = 256$. The brute-force approach is unable to accurately sample values outside of the initial state, and cannot detect bistability in the model.

For the sake of clarity, error-bars were omitted from Fig. 3. Over most of the data range, the error is too small to see on the plot. In the tails (of both SSA and WE-SSA), the error is not computable from a single run, since there are plot points comprised of only a single trajectory. Multiple runs are consistent with the data shown. The error in the estimate of the distribution can be inferred visually from the data’s departure form the CME exact solution. When computing quantitative observables below, we employ multiple independent runs to procure standard errors in our estimate.

WE yields the full, unbiased probability distribution, but we again examine an observable in the spirit of that investigated by Kuwahara and Mura[29] and Gillespie and coworkers,[30–33] i.e., the probability that the progress coordinate is beyond a certain threshold at a specified time, which is a simple summation of the distribution over the state of interest. From ten replicates of the Schlögl run plotted in Fig. 3, the probability that $X \geq 700$ at $t = 5$ s, i.e., $P(X \geq 700, t = 5\,\text{s})$, which we call $p_{\geq 700}$, is $1.143 \times 10^{-9} \pm 4.7 \times 10^{-11}$ at one standard error. The CME exact value is 1.148×10^{-9}. Following Ref. 30, we can estimate the number of brute-force SSA runs that would be needed to find $p_{\geq 700}$ at a similar level of accuracy as $n^{\text{SSA}}_{p_{\geq 700}} = p_{\geq 700}/(\sigma_{p_{\geq 700}})^2 = 1.148 \times 10^{-9}/(4.7 \times 10^{-11})^2 = 5.3 \times 10^{11}$ SSA runs. We can then estimate an improvement in efficiency of using WE over brute-force of $E \sim 5.3 \times 10^{11}/802,000 \sim 7 \times 10^5$ for this observable at this level of accuracy.

We also estimate the mean first passage time (MFPT) of the Schlögl system, which can be computed exactly.[67] Weighted ensemble can estimate the MFPT using Eq. (1) when the system is put into a steady state. For the run that was reweighted every 100 iterations, Fig. 4 shows the WE estimates of the flux from the initial state ($X = 82$) to the final state ($X \geq 563$) converge to the exact value in about 100 iterations of weighted ensemble splittings and mergings, which is when the system relaxes from its delta-function initialization to a steady-state. The attainment of steady state is accelerated by more frequent reweighting (see Sec. II B 2 on reweighting), as is shown in Fig. 4 in the runs that are reweighted every 2 and 5 iterations. These more frequently reweighted runs yield fluxes close to the exact value within about 30 iterations.

To quantify WE’s improvement over brute-force in the estimate of the MFPT, we use the measure $E_{50\%}$ defined in Sec. II C. A brute-force estimate of the MFPT would require, optimistically, computing an amount of dynamics on the order of the MFPT itself (approximately 5×10^4 s). Since transitions in this system follow an exponential distribution, the standard deviation of the first passage times is equal to the mean of them. WE’s estimate of the MFPT is within 50% of the exact value after about 30 iterations of WE simulation, at which point about 1100 trajectory segments have been propagated, which is equivalent to propagating about 110 s of brute-force dynamics. Thus we find $E_{50\%} \sim 5 \times 10^4/110 \approx 500$. As can be seen in Fig. 4, this value is about a 3–5 fold increase over the WE results when reweighting very infrequently (every 100 iterations).

C. Yeast polarization

To extend our comparison to existing methods, we also implemented and benchmarked against a yeast polarization model. This system has been previously studied using different variants of wSSA,[31,33] and presents an opportunity to compare performance gains over brute-force on a small-to-medium sized reaction network of non-trivial complexity.

1. Model

This yeast polarization model[31] consists of six dynamical species (note that in the reactions below, the concentration of L never changes) and eight reactions:

$$\begin{aligned}
&R_1: \quad \varnothing \xrightarrow{k_1} R, && k_1 = 0.0038,\\
&R_2: \quad R \xrightarrow{k_2} \varnothing, && k_2 = 0.0004,\\
&R_3: \quad R + L \xrightarrow{k_3} RL + L, && k_3 = 0.042,\\
&R_4: \quad RL \xrightarrow{k_4} R, && k_4 = 0.010,\\
&R_5: \quad RL + G \xrightarrow{k_5} G_a + G_{bg}, && k_5 = 0.011,\\
&R_6: \quad G_a \xrightarrow{k_6} G_d, && k_6 = 0.100,\\
&R_7: \quad G_{bg} + G_d \xrightarrow{k_7} G, && k_7 = 1050,\\
&R_8: \quad \varnothing \xrightarrow{k_8} RL, && k_8 = 3.21.
\end{aligned} \tag{8}$$

All reaction propensities are in numbers of particles per second. The system is initialized at $t = 0$ with species counts [R, L, RL, G, G_a, G_{bg}, G_d] = [50, 2, 0, 50, 0, 0, 0]. That is, we start with 50 molecules each of R and G, and 2 of L, and no others.

By design, this system does not reach equilibrium;[31] additionally, the rare event measured has no well-defined states, but is merely a measure of an unusually fast accumulation of G_{bg}.

2. WE parameters

The rare event statistics presented below were measured using 50 bins on the interval [0, 49], with an absorbing boundary condition at 50. We used 100 trajectory segments per bin, a time-step $\tau = 0.125$ s, and we run for 160 iterations of that time-step. We note that in situations like these the measurement of rare events with an absorbing boundary condition can be somewhat sensitive to trajectory "bounce-out" from the target state as the time-step is varied. This effect is intrinsic to SSA and not particular to weighted ensemble or other sampling methods.

3. Results

Our measurement of $P(G_{bg} \to 50\,|\,20\text{s})$, i.e., the probability of the population of G_{bg} reaching 50 within 20 s, was $1.20 \times 10^{-6} \pm 0.04 \times 10^{-6}$ at one standard error. We used 100 replicate weighted ensemble runs to estimate the uncertainty.

To check these results, we also performed 80 million brute-force SSA trajectories, which yielded 108 trajectories that successfully reached a G_{bg} population of 50 within 20 s. This brute-force data give an estimate for the rare event probability of $1.35 \times 10^{-6} \pm 0.13 \times 10^{-6}$ at one standard error, which corroborates the weighted ensemble result.

To estimate WE efficiency, again following Gillespie *et al.*,[30] we can estimate the number of brute-force SSA runs that would be needed to find $P(G_{bg} \to 50\,|\,20\text{s})$ at a similar level of accuracy as the weighted ensemble result above: $N_{BF} = P(G_{bg} \to 50\ |\ 20\text{s})/\sigma^2_{P(G_{bg}\to 50|20\text{s})} = 1.20 \times 10^{-6}/(0.04 \times 10^{-6})^2 = 7.3 \times 10^8$. In total, the 100 replicate WE runs in our measurement used 7.205×10^7 trajectory segments, which is equivalent to running $7.205 \times 10^7/160 = 4.5 \times 10^5$ brute force trajectories. This yields a speed-up of WE over brute-force SSA of $E \sim (7.3 \times 10^8)/(4.5 \times 10^5) \sim 1.6 \times 10^3$.

Gillespie and co-workers report values for this same rare event of $1.23 \times 10^6 \pm 0.05 \times 10^6$ and $1.202 \times 10^6 \pm 0.014 \times 10^6$ at two standard errors using two variants of wSSA,[31] with speed-ups over brute-force of 20 and 250, respectively.

D. Epigenetic switch

1. Model

This model consists of two genes that repress each other's expression. Once expressed, each protein can bind particular DNA sites upstream of the gene which codes for the other protein, thereby repressing its transcription.[69] If we denote the ith protein concentration by g_i, the deterministic system is described by the equations:

$$\begin{aligned}
\frac{\mathrm{d}g_1}{\mathrm{d}t} &= \frac{a_1}{1 + (g_2/K_2)^n} - \frac{g_1}{\tau},\\
\frac{\mathrm{d}g_2}{\mathrm{d}t} &= \frac{a_2}{1 + (g_1/K_1)^m} - \frac{g_2}{\tau},
\end{aligned} \tag{9}$$

where $a_1 = 156$, $a_2 = 30$, $n = 3$, $m = 1$, $K_1 = 1$, $K_2 = 1$, and $\tau = 1$. In our stochastic model, our chemical reactions take the form of a birth-death process, the propensity functions of which are taken from the above differential equations:

$$\begin{aligned}
&\varnothing \xrightarrow{k_1(g_2)} g_1 \xrightarrow{k_0} \varnothing,\\
&\varnothing \xrightarrow{k_2(g_1)} g_2 \xrightarrow{k_0} \varnothing,
\end{aligned} \tag{10}$$

where $k_0 = 1/\tau$, $k_1(g_2) = a_1/[1 + (g_2/K_2)^n]$, $k_2(g_1) = a_2/[1 + (g_1/K_1)^m]$.

For this system, we define a target state and an initial configuration. The system is initially set to have $g_1 = 30$ and $g_2 = 0$, and we look for transitions to a target state, which we define as having both $g_1 \leq 20$ and $g_2 \geq 3$. This target state definition was chosen so that the rate was insensitive to small perturbations in the threshold values chosen for g_1 and g_2.

2. WE parameters

For this system, we implemented 2-dimensional bins: 15 along g_1 and 31 along g_2, for a total of 465 bins. The bins along the g_1 coordinate were of unit width on the interval [0, 10], and then of width 10 on the interval [10, 50], with one additional bin on [50, ∞]. The bins along the g_2 coordinate were of unit width on the interval [0, 30], with one additional bin on [30, ∞].

The WE data in Fig. 5 were generated using 16 trajectory segments per bin, a time-step $\tau = 0.1$ s, and run for 500 iterations of that time-step, with reweighting events

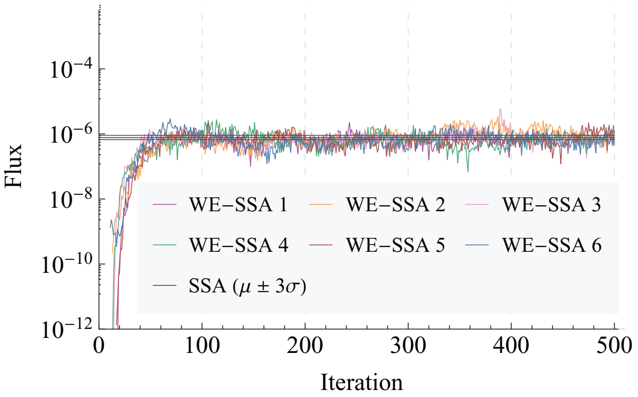

FIG. 5. Measurements of probability flux into the target state for the epigenetic switch system. Six independent WE simulations are plotted, as well as the 3-σ confidence interval for the brute-force data, which are from 753 trajectories of 10^6 s each. The inverse of the flux gives the mean first passage time by Eq. (1).

applied every 100 iterations. Figure 5 shows six independent simulations using these parameters, as well as MLE statistics from our brute-force computations. Were all the bins full at all iterations, WE would compute, for each of the six runs, 3,720,000 trajectory segments of length 0.1 s each, which is equivalent in cost to running 372,000 s of brute-force dynamics. In our case, most of the bins never get populated; we computed dynamics for 148,855, 149,516, 148,940, 147,351, 146,804, and 149,765 segments in the six different runs. In toto, this is equivalent to 89 123.1 s of brute-force dynamics.

3. Results

Even the state-space of this two-species stochastic system is too large to solve exactly, necessitating the use of brute-force simulation as a baseline comparison. A brute-force computation was performed using the SSA as implemented in BNG. 753 simulations of 10^6 s each were run, and using an exponential distribution of MFPTs, the MLE (see Eq. (5)) of the mean and standard error of the mean, $\mu_{\rm MLE}$ and σ_μ, were found to be 1.3×10^6 s and 6.5×10^4 s respectively for transitions from the initial configuration to the target state.

The WE results are plotted against the brute-force values in Fig. 5, where we have used the relation MFPT = 1/flux (Eq. (1)) to plot the steady-state flux that brute-force predicts. We plot the net flux entering the target state as the simulation progresses, because this is what WE measures directly; we can infer the MFPT using the above relation. Taking the mean of each of the six WE runs after the simulation is in steady-state (we discard the first 100 iterations), and treating each of these means as an independent data point, WE gives a combined estimate for the MFPT of $1.3 \times 10^6 \pm 3 \times 10^4$ s at 1-σ for transitions from the initial configuration to the target state.

WE is able to find an estimate of the MFPT with greater precision than brute-force, using the equivalent of 89,123.1 s of brute-force dynamics. The brute-force estimate uses 753×10^6 s of dynamics, yielding a speedup by a factor of $E \sim 10^4$ when using WE compared to brute-force.

WE is also able to quickly attain an efficient rough estimate of the MFPT. A brute-force estimate of the MFPT would require, optimistically, computing an amount of dynamics on the order of the MFPT itself ($\sim 10^6$ s). In the six different simulations, WE's estimate of the MFPT is within 50% of the brute-force value after {52, 44, 37, 40, 43, 42} iterations of WE simulation, at which point {10238, 8400, 6177, 6819, 7141, 7750} trajectory segments have been propagated, which is equivalent to propagating {1023.8, 840.0, 617.7, 681.9, 714.1, 775.0} s of brute-force dynamics, the mean of which is approximately 775. Thus we find a mean $E_{50\%} \approx 1.3 \times 10^6/775 \approx 1725$.

E. FcεRI-mediated signaling

1. Model

To demonstrate the flexibility of the WE approach, we applied it to a signaling model that is, to our knowledge, considerably more complex than any other biochemical system to which rare event sampling techniques have been applied. The reaction network in this model (see the supplementary material[70]) contains 354 chemical species and 3680 chemical reactions.[71]

This model describes association, dissociation, and phosphorylation reactions among four components: the receptor FcεRI, a bivalent ligand that aggregates receptors into dimers, and the protein tyrosine kinases Lyn and Syk. The model also includes dephosphorylation reactions mediated by a pool of protein tyrosine phosphatases. These reactions generate a network of 354 distinct molecular species. The model predicts levels of association and phosphorylation of molecular complexes as they vary with time, ligand concentration, concentrations of signaling components, and genetic modifications of the interacting proteins.

2. WE parameters

The WE data in Fig. 6 were generated using 60 bins of unit width, 100 trajectory segments per bin, a time-step $\tau = 0.6$ s, and run for 100 iterations of that time-step, with no reweighting events. The brute-force data are from 1484 brute-force runs of 60 s each, which is equivalent to the dynamics time employed in attaining a single run of WE data. No attempt was made to optimize sampling times or bin widths in WE.

3. Results

Figure 6 shows the probability distribution of activated receptors in the FcεRI-Mediated Signaling model at time $t = 60$ s. The brute-force SSA approach is unable to sample out to likelihoods much below $\sim 10^{-3}$, while WE gets relatively clean statistics for likelihood values down to $\sim 10^{-8}$, for an estimated improvement in efficiency $E \sim 10^5$.

IV. DISCUSSION

We applied the weighted ensemble (WE)[45–52] approach to systems-biology models of stochastic chemical kinetics

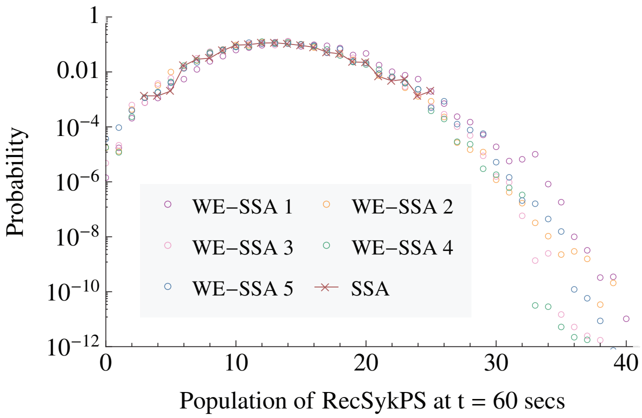

FIG. 6. Comparison of WE and SSA for the FcεRI signaling model, which has 354 reactions and 3680 chemical species. The probability distribution is shown for the system reaching a specified level of Syk activation (the output of the model, which is a sum of 164 species concentrations) within 1 min of system time after stimulation. Results of 1484 SSA simulations of 1 min duration are compared with five independent WE runs, each generated with an equivalent computational effort as that of the brute-force SSA (several CPU h in each case).

equations, implemented in BioNetGen.[17,54] Increases in computational efficiency on the order of 10^{18} were attained for a simple system of biological relevance (the enzymatic futile cycle), and on the order of 10^5 for a large systems-biology model (FcεRI), with 354 species and 3680 reactions.

WE is easy to understand and implement, statistically exact,[47] and easy to parallelize. It can yield long-timescale information such as mean first passage times (MFPTs) from simulations of much shorter length. As in prior molecular simulations,[46,48] WE has been demonstrated to increase computational efficiency by orders of magnitude for models of non-trivial complexity, and offers perfect (linear) parallel scaling. It appears that WE holds significant promise as a tool for the investigation of complex stochastic systems.

Nevertheless, a number of additional points, including limitations of WE and related procedures, merit further discussion.

A. Strengths of WE

Beyond the efficiency observed for the systems studied here, the WE approach has other significant strengths. Weighted ensemble is easy to implement: it examines trajectories at fixed time-intervals, and its implementation as scripting-level code makes it amenable to using any stochastic dynamics engine to propagate trajectories. WE also parallelizes well, and can take advantage of multiple cores on a single machine, or across many machines on a cluster; Zwier *et al.* have successfully performed a WE computation on more than 2000 cores on the Ranger supercomputer.[52] Additionally, WE trajectories are unbiased in the sense that they always follow the natural dynamics of the system — there is no need to engage in the potentially quite challenging process of adjusting the internal dynamics of the simulation to encourage rare events. WE also yields full probability distributions, and can find mean first passage times (MFPTs) and equilibrium properties of systems.

Is there a curse of dimensionality? In other words, for successful WE, does one need to know (and cut into bins) all slow coordinates? The short answer is that one only needs to sub-divide independent/uncorrelated slow coordinates: other, correlated coordinates, by definition, come along for the ride. Recall that WE does not apply forces to coordinates, but only “replicates success.” As evidenced by the FcεRI model in Sec. III E, WE can accurately and efficiently sample rare events for a system of hundreds of degrees of freedom by binning on only the coordinate of interest, without worrying about any intermediate degrees of freedom.

For molecular systems, conventional biophysical wisdom holds that functional transitions are dominated by relatively low-dimensional “tubes” of configuration space.[72,73] In colloquial terms, for biological systems to function in real time, there is a limit to the “functional dimensionality” of the configuration space that can be explored, echoing the resolution of Levinthal’s paradox.[74,75] Nevertheless, the ultimate answer to the issue of dimensionality in systems biology must await more exploration and may well be system dependent.

B. Comparison to other approaches

WE is most similar in spirit to recent versions of forward flux sampling (FFS)[36–38] and non-equilibrium umbrella sampling (NEUS).[39] All of these methods divide up state-space into different regions, and are able to merge and split trajectories so as to enhance the sampling of rare regions of state-space. These approaches differ slightly in the way the splitting and merging of trajectories is performed. WE also differs from FFS in that WE does not have to catch trajectories in the act of crossing a bin boundary; instead WE checks, at a prescribed time step, in which bin a trajectory resides. Though WE and FFS should be expected to perform comparably, this subtle difference can be advantageous in that no low-level interaction with the dynamics engine/software is required in WE, which makes an implementation of the WE algorithm more flexible and easy to apply to diverse systems.

The central hurdle to improving efficiency using accelerating sampling techniques such as WE, FFS, and NEUS, is to adequately divide that state-space by selecting reaction coordinates that are both important to the dynamics of interest, and that are slowly sampled by brute-force approaches. Optimally and automatically dividing and binning the state-space is, to our knowledge, an open problem, and one that, for complex systems, where a target state is unknown, is not always a straightforward one to solve, though adaptive strategies have been suggested.[47,50,76]

The wSSA method[29] differs from the above procedures. It does not use a state-space approach, but rather uses importance sampling to bias and then unbias the reaction rates in a manner that yields an unbiased estimation of the desired observable. WE exhibits comparable or better performance to wSSA for systems to which both can be applied. Of the two models we ran for side-by-side comparisons, wSSA outperformed WE by a factor of about 100 for the simple futile-cycle model, while WE outperformed wSSA by a factor of about 6–7 on the somewhat more complex yeast polarization model. Since wSSA biases/unbiases reaction rates, while WE divides state-space, the advantage of one over the other may be situation-dependent. For instance, as noted by the wSSA authors,[30] when a reaction network is finely tuned and exhibits qualitatively different behavior outside a narrow range of parameter space (e.g., the Schlögl reactions), employing a strategy that changes the rates of reactions can be very challenging. The ease of implementation of the WE framework, which does not require the potentially challenging task of biasing and reweighting multi-dimensional dynamics, would appear to scale better with model complexity than current versions of wSSA; however, for very small models such as the futile cycle, wSSA may outperform WE in measuring select observables.

A limitation which would appear to be common to accelerated sampling techniques employed to estimate non-equilibrium observables is the system-intrinsic timescale: "t_b", or the "event duration" time.[77,78] This timescale represents the time it takes for realistic (unbiased) trajectories to "walk" from one state to another, excluding the waiting time prior to the event. The event duration is often only a fraction of the MFPT, since it is the *likelihood* of walking this path that is low; the time to actually walk the path is often quite moderate. WE excels at overcoming the low likelihood of a transition, but it would appear difficult if not impossible for any technique which generates transition trajectories to overcome t_b, which is the intrinsic timescale of transition events.

Finally, it should be noted that all state-space methods that branch trajectories, including WE, typically produce correlated trajectories, due to the splitting/merging events. For example, the presence of 10 trajectories in a bin does not imply 10 statistically independent samples. While such correlations do not appear to have impeded the application of WE to the systems investigated here, future work will aim to quantify their effects and reduce their potential impact. The present work accounted for correlations by analyzing multiple fully independent WE runs.

C. Future applications

Beyond potential applications to more complex stochastic chemical kinetics models, the weighted ensemble formalism could be applied to spatially heterogeneous systems. WE should be able to accelerate the sampling of models such as those generated by MCell[79–81] or Smoldyn,[82] perhaps using three-dimensional spatial bins.

It may be possible to integrate WE with other methods. We note that the state-space dividing approaches of a number of methods (forward flux,[20,34–36] non-equilibrium umbrella sampling,[39,40] and weighted ensemble[45–52]), since they are dynamics-agnostic, could be combined with other methods that accelerate the dynamics engine itself, such as the τ-leaping modification of Gillespie's SSA and its many variants and improvements,[83–86] to yield multiplicative increases in runtime speedup.

More speculatively, WE could be combined with parallel tempering methods.[87–89] WE accelerates the exploration of the free-energy landscape at a given temperature, and since it does not bias dynamics, the trajectories it propagates could be suitable for replica exchange schemes.

For complex models where exploring the state-space via brute-force is prohibitively expensive, WE could also be employed to search for bistability, or in a model-checking capacity[90–92] to search for pathological states.

ACKNOWLEDGMENTS

We gratefully acknowledge funding from NSF Grant No. MCB-1119091, NIH Grant No. P41 GM103712, NIH Grant No. T32 EB009403, and NSF Expeditions in Computing Grant (Award No. 0926181). We thank Steve Lettieri, Ernesto Suarez, and Justin Hogg for helpful discussions.

[1] M. Esteller, N. Engl. J. Med. **358**, 1148 (2008).
[2] H. H. McAdams and A. Arkin, Proc. Natl. Acad. Sci. U.S.A. **94**, 814 (1997).
[3] A. Arkin, J. Ross, and H. H. McAdams, Genetics **149**, 1633 (1998).
[4] W. J. Blake, M. Kaern, C. R. Cantor, and J. J. Collins, Nature (London) **422**, 633 (2003).
[5] J. M. Raser and E. K. O'Shea, Science **304**, 1811 (2004).
[6] L. S. Weinberger, J. C. Burnett, J. E. Toettcher, A. P. Arkin, and D. V. Schaffer, Cell **122**, 169 (2005).
[7] M. Acar, J. T. Mettetal, and A. van Oudenaarden, Nat. Genet. **40**, 471 (2008).
[8] L. Cai, N. Friedman, and X. S. Xie, Nature (London) **440**, 358 (2006).
[9] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, Science **297**, 1183 (2002).
[10] A. Raj and A. van Oudenaarden, Cell **135**, 216 (2008).
[11] N. Maheshri and E. K. O'Shea, Annu. Rev. Biophys. Biomol. Struct. **36**, 413 (2007).
[12] B. B. Kaufmann and A. van Oudenaarden, Curr. Opin. Genet. Dev. **17**, 107 (2007).
[13] V. Shahrezaei and P. S. Swain, Curr. Opin. Biotechnol. **19**, 369 (2008).
[14] A. Raj and A. van Oudenaarden, Annu. Rev. Biophys. **38**, 255 (2009).
[15] D. T. Gillespie, Annu. Rev. Phys. Chem. **58**, 35 (2007).
[16] D. J. Wilkinson, *Stochastic Modelling for Systems Biology*, 2nd ed. (CRC Press, 2011), p. 335.
[17] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek, Bioinformatics **20**, 3289 (2004).
[18] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[19] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[20] R. Allen, P. Warren, and P. ten Wolde, Phys. Rev. Lett. **94**, 018104 (2005).
[21] W. Zhou, X. Peng, Z. Yan, and Y. Wang, Comput. Biol. Chem. **32**, 240 (2008).
[22] E. Mjolsness, D. Orendorff, P. Chatelain, and P. Koumoutsakos, J. Chem. Phys. **130**, 144110 (2009).
[23] D. D. Jenkins and G. D. Peterson, Comput. Phys. Commun. **182**, 2580 (2011).
[24] A. Chatterjee, K. Mayawala, J. S. Edwards, and D. G. Vlachos, Bioinformatics **21**, 2136 (2005).
[25] B. Bayati, P. Chatelain, and P. Koumoutsakos, J. Comput. Phys. **228**, 5908 (2009).
[26] D. T. Gillespie and L. R. Petzold, J. Chem. Phys. **119**, 8229 (2003).
[27] H. Lu and P. Li, Comput. Phys. Commun. **183**, 1427 (2012).
[28] M. A. Gibson and J. Bruck, J. Phys. Chem. A **104**, 1876 (2000).
[29] H. Kuwahara and I. Mura, J. Chem. Phys. **129**, 165101 (2008).
[30] D. T. Gillespie, M. Roh, and L. R. Petzold, J. Chem. Phys. **130**, 174103 (2009).
[31] M. K. Roh, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **133**, 174106 (2010).
[32] B. J. Daigle, M. K. Roh, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **134**, 044110 (2011).
[33] M. K. Roh, B. J. Daigle, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **135**, 234108 (2011).
[34] R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. **124**, 024102 (2006).

[35] R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. **124**, 194111 (2006).

[36] R. J. Allen, C. Valeriani, and P. R. ten Wolde, J. Phys.: Condens. Matter **21**, 463102 (2009).

[37] N. B. Becker, R. J. Allen, and P. R. ten Wolde, J. Chem. Phys. **136**, 174118 (2012).

[38] N. B. Becker and P. R. ten Wolde, J. Chem. Phys. **136**, 174119 (2012).

[39] A. Dickson, A. Warmflash, and A. R. Dinner, J. Chem. Phys. **130**, 074104 (2009).

[40] A. Warmflash, P. Bhimalapuram, and A. R. Dinner, J. Chem. Phys. **127**, 154112 (2007).

[41] D. M. Zuckerman and T. B. Woolf, J. Chem. Phys. **111**, 9475 (1999).

[42] A. K. Faradjian and R. Elber, J. Chem. Phys. **120**, 10880 (2004).

[43] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, J. Chem. Phys. **108**, 1964 (1998).

[44] T. S. van Erp, D. Moroni, and P. G. Bolhuis, J. Chem. Phys. **118**, 7762 (2003).

[45] G. A. Huber and S. Kim, Biophys. J. **70**, 97 (1996).

[46] B. W. Zhang, D. Jasnow, and D. M. Zuckerman, Proc. Natl. Acad. Sci. U.S.A. **104**, 18043 (2007).

[47] B. W. Zhang, D. Jasnow, and D. M. Zuckerman, J. Chem. Phys. **132**, 054107 (2010).

[48] D. Bhatt, B. W. Zhang, and D. M. Zuckerman, J. Chem. Phys. **133**, 014110 (2010).

[49] M. C. Zwier, J. W. Kaus, and L. T. Chong, J. Chem. Theory Comput. **7**, 1189 (2011).

[50] J. L. Adelman and M. Grabe, J. Chem. Phys. **138**, 044105 (2013).

[51] S. Lettieri, M. C. Zwier, C. A. Stringer, E. Suarez, L. T. Chong, and D. M. Zuckerman, "Simultaneous computation of dynamical and equilibrium information using a weighted ensemble of trajectories," e-print arXiv:1210.3094.

[52] M. Zwier, J. Adelman, J. Kaus, S. Lettieri, E. Suarez, D. Wang, D. Zuckerman, M. Grabe, and L. Chong, "WESTPA: A portable, highly scalable software package for weighted ensemble simulation and analysis, (unpublished).

[53] M. C. Zwier and L. T. Chong, Curr. Opin. Pharmacol. **10**, 745 (2010).

[54] J. R. Faeder, M. L. Blinov, and W. S. Hlavacek, Methods Mol. Biol. **500**, 113 (2009).

[55] A. Nag, M. I. Monine, J. R. Faeder, and B. Goldstein, Biophys. J. **96**, 2604 (2009).

[56] O. Feinerman, G. Jentsch, K. E. Tkach, J. W. Coward, M. M. Hathorn, M. W. Sneddon, T. Emonet, K. A. Smith, and G. Altan-Bonnet, Mol. Syst. Biol. **6**, 437 (2010).

[57] H. Gong, P. Zuliani, A. Komuravelli, J. R. Faeder, and E. M. Clarke, BMC Bioinf. **11**(Suppl. 7), S10 (2010).

[58] T. M. Thomson, K. R. Benjamin, A. Bush, T. Love, D. Pincus, O. Resnekov, R. C. Yu, A. Gordon, A. Colman-Lerner, D. Endy, and R. Brent, Proc. Natl. Acad. Sci. U.S.A. **108**, 20265 (2011).

[59] D. Barua, W. S. Hlavacek, and T. Lipniacki, J. Immunol. **189**, 646 (2012).

[60] T. L. Hill, *Free Energy Transduction and Biochemical Cycle Kinetics* (Dover Publications, 2004), p. 119.

[61] B. Zagrovic and V. Pande, J. Comput. Chem. **24**, 1432 (2003).

[62] M. Samoilov, S. Plyasunov, and A. P. Arkin, Proc. Natl. Acad. Sci. U.S.A. **102**, 2310 (2005).

[63] A. Warmflash, D. N. Adamson, and A. R. Dinner, J. Chem. Phys. **128**, 225101 (2008).

[64] B. N. Kholodenko, Nat. Rev. Mol. Cell Biol. **7**, 165 (2006).

[65] L. Wang and E. D. Sontag, J. Math. Biol. **57**, 29 (2008).

[66] F. Schlögl, Z. Phys. **253**, 147 (1972).

[67] D. T. Gillespie, *Markov Processes: An Introduction for Physical Scientists* (Gulf Professional Publishing, 1992), p. 565.

[68] M. Vellela and H. Qian, J. R. Soc., Interface **6**, 925 (2009).

[69] D. Roma, R. O'Flanagan, A. Ruckenstein, A. Sengupta, and R. Mukhopadhyay, Phys. Rev. E **71**, 011902 (2005).

[70] See supplementary material at http://dx.doi.org/10.1063/1.4821167 for details of the model.

[71] J. R. Faeder, W. S. Hlavacek, I. Reischl, M. L. Blinov, H. Metzger, A. Redondo, C. Wofsy, and B. Goldstein, J. Immunol. **170**, 3769 (2003).

[72] W. E, W. Ren, and E. Vanden-Eijnden, Chem. Phys. Lett. **413**, 242 (2005).

[73] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, Annu. Rev. Phys. Chem. **53**, 291 (2002).

[74] A. Sali, E. Shakhnovich, and M. Karplus, Nature (London) **369**, 248 (1994).

[75] K. A. Dill and H. S. Chan, Nat. Struct. Biol. **4**, 10 (1997).

[76] D. Bhatt and I. Bahar, J. Chem. Phys. **137**, 104101 (2012).

[77] D. M. Zuckerman and T. B. Woolf, J. Chem. Phys. **116**, 2586 (2002).

[78] B. W. Zhang, D. Jasnow, and D. M. Zuckerman, J. Chem. Phys. **126**, 074504 (2007).

[79] J. Stiles, and T. Bartol, in *Computational Neuroscience: Realistic Modeling for Experimentalists*, edited by E. De Schutter (CRC Press, Boca Raton, 2001), pp. 87–127.

[80] R. A. Kerr, T. M. Bartol, B. Kaminsky, M. Dittrich, J.-C. J. Chang, S. B. Baden, T. J. Sejnowski, and J. R. Stiles, SIAM J. Sci. Comput. (USA) **30**, 3126 (2008).

[81] J. R. Stiles, D. Van Helden, T. M. Bartol, E. E. Salpeter, and M. M. Salpeter, Proc. Natl. Acad. Sci. U.S.A. **93**, 5747 (1996).

[82] S. S. Andrews, Methods Mol. Biol. **804**, 519 (2012).

[83] L. A. Harris and P. Clancy, J. Chem. Phys. **125**, 144107 (2006).

[84] T. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004).

[85] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **124**, 044109 (2006).

[86] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).

[87] U. H. Hansmann, Chem. Phys. Lett. **281**, 140 (1997).

[88] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).

[89] D. J. Earl and M. W. Deem, Phys. Chem. Chem. Phys. **7**, 3910 (2005).

[90] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Cheking* (MIT Press, 1999), p. 314.

[91] E. M. Clarke, J. R. Faeder, C. J. Langmead, L. A. Harris, S. K. Jha, and A. Legay, *Proceedings of the 6th Annual Conference on Computational Methods in Systems Biology*, Lecture Notes in Computer Science (Bioinformatics) Vol. 5307 (Springer-Verlag, 2008), pp. 231–250.

[92] S. K. Jha, E. M. Clarke, C. J. Langmead, A. Legay, A. Platzer, and P. Zuliani, *Proceedings of the 7th Annual Conference on Computational Methods in Systems Biology*, Lecture Notes in Computer Science (Bioinformatics) Vol. 5688 (Springer-Verlag, 2009), pp. 218–234.