Variable-free exploration of stochastic models: A gene regulatory network example

Radek Erban[a)]
Mathematical Institute, University of Oxford, 24-29 St. Giles', Oxford OX1 3LB, United Kingdom

Thomas A. Frewen[b)]
Department of Chemical Engineering, Princeton University, Princeton, New Jersey 08544

Xiao Wang[c)]
Department of Statistics and Operations Research, Bioinformatics and Computational Biology Program, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27599

Timothy C. Elston[d)]
Department of Pharmacology, University of North Carolina, Chapel Hill, North Carolina 27599

Ronald Coifman[e)]
Department of Mathematics, Yale University, New Haven, Connecticut 06520

Boaz Nadler[f)]
Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel

Ioannis G. Kevrekidis[g)]
Department of Chemical Engineering, PACM and Mathematics, Princeton University, Princeton, New Jersey 08544

(Received 6 November 2006; accepted 27 February 2007; published online 19 April 2007)

Finding coarse-grained, low-dimensional descriptions is an important task in the analysis of complex, stochastic models of gene regulatory networks. This task involves (a) identifying observables that best describe the state of these complex systems and (b) characterizing the dynamics of the observables. In a previous paper [R. Erban *et al.*, J. Chem. Phys. **124**, 084106 (2006)] the authors assumed that good observables were known *a priori*, and presented an equation-free approach to approximate coarse-grained quantities (i.e., effective drift and diffusion coefficients) that characterize the long-time behavior of the observables. Here we use diffusion maps [R. Coifman *et al.*, Proc. Natl. Acad. Sci. U.S.A. **102**, 7426 (2005)] to extract appropriate observables ("reduction coordinates") in an *automated* fashion; these involve the leading eigenvectors of a weighted Laplacian on a graph constructed from network simulation data. We present lifting and restriction procedures for translating between physical variables and these data-based observables. These procedures allow us to perform equation-free, coarse-grained computations characterizing the long-term dynamics through the design and processing of short bursts of stochastic simulation initialized at appropriate values of the data-based observables. © *2007 American Institute of Physics*. [DOI: 10.1063/1.2718529]

I. INTRODUCTION

Gene regulatory networks are complex high-dimensional stochastic dynamical systems. These systems are subject to large intrinsic fluctuations that arise from the inherent random nature of the biochemical reactions that constitute the network. Such features make realistic modeling of genetic networks, based on exact representations of the chemical master equation [such as the Gillespie stochastic simulation algorithm[1] (SSA)] computationally expensive. Recently there has been considerable work devoted to developing efficient numerical algorithms for accelerating the stochastic simulation of gene regulatory networks[2–5] and, more generally, of chemical reaction networks. Many of these techniques are based on time scale separation and classify the biochemical reactions as "slow" or "fast".[6–10] In this paper we combine such acceleration methods with recently developed data-mining techniques (in particular, diffusion maps[11–13]) capable of identifying appropriate coarse-grained variables ("observables" and "reduction coordinates") based on simulation data. These observables are then used in the context of accelerating stochastic gene regulatory network simulations; they guide the design, initialization, and processing of the results of short bursts of full-scale SSA computation. These bursts of SSA are used to numerically solve the (unavailable in closed form) evolution equations for the

[a)]Electronic mail: erban@maths.ox.ac.uk
[b)]Electronic mail: tfrewen@princeton.edu
[c)]Electronic mail: xiaow@email.unc.edu
[d)]Electronic mail: telston@amath.unc.edu
[e)]Electronic mail: coifman-ronald@yale.edu
[f)]Electronic mail: boaz.nadler@weizmann.ac.il
[g)]Electronic mail: yannis@princeton.edu

observables; such so-called equation-free methods[14] for studying stochastic models have been successfully applied to complex systems arising in different contexts.[15–17] In the context of gene regulatory networks—but with *known* observables—equation-free modeling has been illustrated in Ref. 4; here we extend the approach to the more general class of problems where appropriate observables are unknown *a priori*.

We describe the state of a gene regulatory network through a vector

$$\mathbf{X} = [X_1, X_2, X_3, \ldots, X_N], \tag{1.1}$$

where X_i are the numbers of various protein molecules, RNA molecules, and genes in the system. The behavior of the gene regulatory network is described by the time evolution of the vector $\mathbf{X}(t)$. For naturally occurring gene regulatory networks the dimension N of the vector $\mathbf{X}(t)$ is, in general, moderately large, ranging from tens to hundreds of species. However, the temporal evolution of the network over time scales of interest can be often usefully described by a much smaller number n of coordinates. For example, in Ref. 4, we studied various models of a genetic toggle switch with $N=2$, $N=4$, and $N=6$ components of the vector $\mathbf{X}$; yet in all cases, the slow dynamics was effectively one dimensional, and a single linear combination of protein concentrations was sufficient to describe the system, i.e., $n=1$. In this paper we show how, for this genetic network system, good coarse variables can be found by data-mining-type methods based on the diffusion map approach.

This paper is organized as follows: We begin with a brief description of our model in Sec. II. Section III quickly reviews the equation-free approach for this type of bistable dynamics. Given a low-dimensional set of observables, the main idea is to locally estimate drift and diffusion coefficients of an unavailable Fokker-Planck equation in these observables from short bursts of appropriately initialized full stochastic simulations. In Sec. IV we show how to process the data generated by stochastic simulations to obtain data-driven observables through the construction of diffusion maps.[13,18] The leading eigenvectors of the weighted graph Laplacian defined on a graph based on simulation data suggest appropriate "automated" reduction coordinates when these are not known *a priori*. Such observables are then used to perform "variable-free" computations. In Sec. V we present lifting and restriction procedures for translating between physical system variables and the automated observables. The bursts of stochastic simulation required for equation-free numerics are designed (and processed) based on these new coordinates. This combined "variable-free, equation-free" analysis appears to be a promising approach for computing features of the long-time, coarse-grained behavior of certain classes of complex stochastic models (in particular, models of gene regulatory networks), as an alternative to long, full SSA simulations. The approach can, in principle, also be wrapped around different types of full atomistic/stochastic simulators, beyond SSA, and, in particular, accelerated SSA approaches such as implicit tau leaping[19] or nested SSA.[9,20]

II. MODEL DESCRIPTION

Our illustrative example is a two-gene network in which each protein represses the transcription of the other gene (mutual repression). This type of system has been engineered in *E. coli* and is often referred to as a genetic toggle switch.[21,22] The advantage of this simple system is that it allows us to test the accuracy of computational methods by direct comparison with results from long-time stochastic simulations. More details about the model can be found in Ref. 23 and in our previous paper.[4] The system contains two genes with operators O_1 and O_2, two proteins P_1 and P_2, and the corresponding dimers, i.e., $N=6$ in Eq. (1.1). The production of P_1 (P_2) depends on the chemical state of the upstream operator O_1 (O_2). If O_1 is empty then P_1 is produced at the rate γ_1 and if O_1 is occupied by a dimer of P_2, then protein P_1 is produced at a rate $\varepsilon_1 < \gamma_1$. Similarly, if O_2 is empty then P_2 is produced at the rate γ_2 and if O_2 is occupied by a dimer of P_1, then protein P_2 is produced at a rate $\varepsilon_2 < \gamma_2$. Note that, for simplicity, transcription and translation are described by a single rate constant. The biochemical reactions are (compare with Ref. 4)

$$\emptyset \underset{\delta_1}{\overset{\gamma_1 O_1 + \varepsilon_1 \overline{P_2P_2O_1}}{\rightleftarrows}} P_1, \quad \emptyset \underset{\delta_2}{\overset{\gamma_2 O_2 + \varepsilon_2 \overline{P_1P_1O_2}}{\rightleftarrows}} P_2, \tag{2.1}$$

$$P_1 + P_1 \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} \overline{P_1P_1}, \quad P_2 + P_2 \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} \overline{P_2P_2}, \tag{2.2}$$

$$\overline{P_2P_2} + O_1 \underset{k_{-o1}}{\overset{k_{o1}}{\rightleftarrows}} \overline{P_2P_2O_1}, \quad \overline{P_1P_1} + O_2 \underset{k_{-o2}}{\overset{k_{o2}}{\rightleftarrows}} \overline{P_1P_1O_2}, \tag{2.3}$$

where overbars denote complexes. Equations (2.1) describe production and degradation of proteins P_1 and P_2, Eqs. (2.2) are dimerization reactions, and Eqs. (2.3) represent the binding and dissociation of the dimer and DNA.

The state vector for our system is

$$\mathbf{X} = [P_1, P_2, \overline{P_1P_1}, \overline{P_2P_2}, O_1, O_2], \tag{2.4}$$

where P_1 and P_2 are numbers of proteins, $\overline{P_1P_1}$ and $\overline{P_2P_2}$ are numbers of dimers, and $O_1 \in \{0,1\}$ and $O_2 \in \{0,1\}$ are states of operators. Assuming that we have just one copy of gene 1 and one copy of gene 2 in the system, then the values of O_1 and $\overline{P_2P_2O_1}$, respectively, O_2 and $\overline{P_1P_1O_2}$, are related by the conservation relations, namely,

$$\overline{P_2P_2O_1} = 1 - O_1, \quad \overline{P_1P_1O_2} = 1 - O_2.$$

By virtue of Eq. (2.1), $O_1=1$ means that the first protein is produced with rate γ_1, while $O_1=0$ means that it is produced with rate $\varepsilon_1 < \gamma_1$ (similarly for the second protein).

Models such as the one defined by Eqs. (2.1)–(2.3) can be validated experimentally, by comparing their predictions with steady-state distributions of protein abundances obtained through single cell fluorescence measurements of intercellular variability in protein expression levels.

III. BRIEF REVIEW OF EQUATION-FREE COMPUTATIONS

Suppose we have a well-stirred mixture of N chemically reacting species; furthermore, assume that the evolution of the system can be described in terms of $n<N$ slow variables (observables). In the following we assume that $n=1$, and denote this variable Q. The approach carries through for the case of a relatively small number of slow variables as well. The variable Q might be the concentration of one of the chemical species or some function of these concentrations (e.g., a linear combination of some of them). In Sec. IV A we show how variable-free methods can be used to suggest an appropriate Q. Let $\mathbf{R}$ denote a vector of the remaining (fast, "slaved") system observables which, together with Q, provide a basis for the simulation space. Our assumption implies that (possibly, after a short initial transient) the evolution of the system can be approximately described by the time-dependent probability density function $f(q,t)$ for the slow variable Q that evolves according to the following *effective* Fokker-Planck equation:[24]

$$\frac{\partial f}{\partial t}(q,t) = \frac{\partial}{\partial q}\left(-V(q)f(q,t) + \frac{\partial}{\partial q}[D(q)f(q,t)]\right). \tag{3.1}$$

If the effective drift $V(q)$ and the effective diffusion coefficient $D(q)$ are explicitly known functions of q, then Eq. (3.1) can be used to compute interesting long-time properties of the system (e.g., the equilibrium distribution and transition times between metastable states). Assuming that Eq. (3.1) provides a good approximation,[21,23] and motivated by the formulas

$$V(q) = \lim_{\Delta t \to 0} \frac{\langle Q(t+\Delta t) - q | Q(t) = q \rangle}{\Delta t}, \tag{3.2}$$

$$D(q) = \frac{1}{2} \lim_{\Delta t \to 0} \frac{\langle [Q(t+\Delta t) - q]^2 | Q(t) = q \rangle}{\Delta t}, \tag{3.3}$$

we used in Refs. 15–17 and 25 the results of short δ-function initialized simulation bursts to estimate the average drift, V, and diffusion coefficient D. Note that, in our context, the limit $\Delta t \to 0$ in Eqs. (3.2) and (3.3) should be interpreted as "Δt small, but not too small," i.e., the short bursts are short in the time scale of the slow variable, yet long in comparison to the characteristic equilibration time of the remaining system variables.

The equilibrium solution of Eq. (3.1) is proportional to $\exp[-\beta\Phi(q)]$, where the effective free energy $\Phi(q)$ is defined as

$$\beta\Phi(q) = -\int_0^q \frac{V(q')}{D(q')} dq' + \ln D(q) + \text{const}. \tag{3.4}$$

Consequently, computing the effective free energy and the equilibrium probability distribution can be accomplished without the need for long-time stochastic simulations. A procedure for computationally estimating $V(q)$ and $D(q)$ is as follows:

(A) Given $Q=q$, approximate the conditional density $P(\mathbf{r}|Q=q)$ for the fast variables $\mathbf{R}$. Details of this preparatory step were given in Ref. 4.
(B) Use $P(\mathbf{r}|Q=q)$ from step (A) to determine appropriate initial conditions for the short simulation bursts and run multiple realizations for time Δt. Use the results of these simulations and the formulae (3.2) and (3.3) to estimate the effective drift $V(q)$ and the effective diffusion coefficient $D(q)$.
(C) Repeat steps (A) and (B) for sufficiently many values of Q and then compute $\Phi(q)$ using formula (3.4) and numerical quadrature.

Determining the accuracy of these estimates and, in particular, the number of replica simulations required for a prescribed accuracy, is the subject of current work. An important feature of this algorithm is that it is trivially parallelizable (different realizations of short simulations starting at "the same q" as well as realizations starting at different q values can be run independently, on multiple processors).

A representative selection of equation-free results from our previous paper,[4] for a stochastic model of a gene regulatory network, is provided in Fig. 1. In Ref. 4 the (good) observable Q was assumed to be known *a priori*. The upper left panel in Fig. 1 shows a sample time series of Q, clearly indicative of bistability, generated using the stochastic model, while the upper right panel shows the effective free energy $\beta\Phi$ computed using Eq. (3.4) as the parameter $\gamma \equiv \gamma_1 = \gamma_2$ is varied. The equation-free steady-state distribution of Q obtained from this effective free energy is in excellent agreement with histograms produced using long-time simulation (lower left panel). Equation-free computation has also been used[4] to compute "stochastic bifurcation diagrams" (an example is shown in the bottom right panel of Fig. 1) using an extension of deterministic bifurcation computation.[26] We believe that this array of equation-free numerical techniques holds promise for the acceleration of computer-assisted analysis of gene regulatory networks. We now extend this analysis to systems where the "good" observables are unknown *a priori* by describing diffusion map based variable-free methods.

IV. VARIABLE-FREE METHODS

A. Theoretical framework

To find a good, low (n-)dimensional representation of the full N-dimensional stochastic simulation data, we start by exploring the phase space of most likely configurations of the system through extensive stochastic simulations; these configurations $\mathbf{X}$ (or a representative sampling of them) at, say, M different times are stored for processing. From M such recordings we obtain a set of M vectors $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(M)}$ in $\mathbb{R}^N$ which constitute the input to the diffusion map dimensionality reduction approach we will now describe. A crucial step for dimensionality reduction is the definition of a meaningful *local* distance measure between configurations. For continuous systems with equal noise strengths in all variables, one may use the following pairwise similarity matrix:

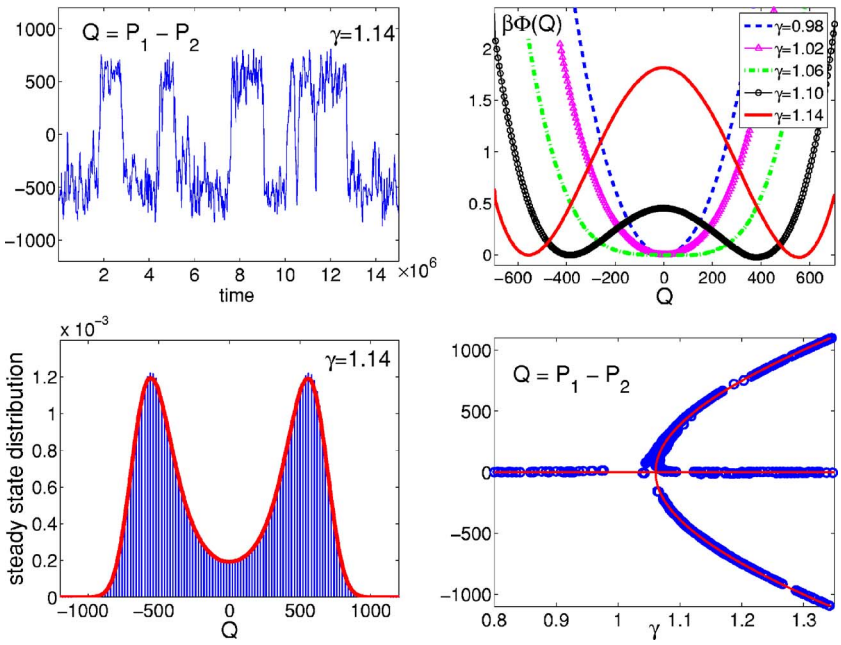

FIG. 1. (Color online) Summary of equation-free results from Ref. 4. To compute the figures we used models (2.1)–(2.3) where Eqs. (2.2) and (2.3) were assumed to be at quasi-equilibrium; for parameter values, see caption of Fig. 5 in Ref. 4.

$$\tilde{W}_{ij} = \exp\left[-\left(\frac{\|\mathbf{X}^{(i)} - \mathbf{X}^{(j)}\|}{\sigma}\right)^2\right], \tag{4.1}$$

where $\|\cdot\|$ is the standard Euclidean norm in $\mathbb{R}^N$ and σ is a characteristic scale for the exponential kernel which quantifies the "locality" of the neighborhood in which the Euclidean distance is considered (dynamically) meaningful.[11]

For discrete chemical and biological reactions, as well as in other systems where the components of the data vectors may be disparate quantities varying over different orders of magnitude (possibly including even Boolean variables), the simple Euclidean norm in Eq. (4.1) with a single scaling factor σ equal for all components may, of course, not be appropriate. In this case, it is reasonable to consider different scalings for the N different components, using an N-dimensional weight vector

$$\mathbf{a} = [a_1, a_2, \ldots, a_N], \tag{4.2}$$

where $a_i > 0$, for $i = 1, \ldots, N$, and define a *weighted* Euclidean norm

$$\|\mathbf{X}\|_{\mathbf{a}}^2 = \sum_{j=1}^{N} (a_j X_j)^2. \tag{4.3}$$

This norm replaces the standard Euclidean norm in Eq. (4.1), where we may now choose $\sigma = 1$, since this scaling can be absorbed into the vector $\mathbf{a}$; thus we replace Eq. (4.1) by

$$\tilde{W}_{ij} = \exp[-\|\mathbf{X}^{(i)} - \mathbf{X}^{(j)}\|_{\mathbf{a}}^2]. \tag{4.4}$$

The elements of the matrix $\tilde{\mathbf{W}}$ are all less than or equal to 1. Nearby points have $\tilde{W}_{ij}$ close to 1, whereas distant points have $\tilde{W}_{ij}$ close to 0. In the diffusion map approach, given $\alpha \in [0,1]$ (the choice of this parameter value is discussed later), we define the matrix $\mathbf{W}$ by

$$W_{ij} = \left(\sum_{k=1}^{M} \tilde{W}_{ik}\right)^{-\alpha} \left(\sum_{k=1}^{M} \tilde{W}_{jk}\right)^{-\alpha} \tilde{W}_{ij}. \tag{4.5}$$

Next, we define a diagonal $M \times M$ normalization matrix $\mathbf{D}$ whose values are given by

$$D_{ii} = \sum_{k=1}^{M} W_{ik}. \tag{4.6}$$

Finally, we compute the eigenvalues and right eigenvectors of the matrix

$$\mathbf{K} = \mathbf{D}^{-1}\mathbf{W}. \tag{4.7}$$

In this paper we will mainly work with the parameter $\alpha = 0$. However, in other applications different values of α may be more suitable (see Appendix A). As discussed in Refs. 13, 18, and 27, if there exists a *spectral gap* among the eigenvalues of this matrix, then the leading eigenvectors may be used as a basis for a low-dimensional representation of the data (see Appendix A). To compute these eigenvectors, we can make use of the fact that

$$\mathbf{K} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{1/2} \quad \text{where } \mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \tag{4.8}$$

is a symmetric matrix. Hence, $\mathbf{K}$ and $\mathbf{S}$ are similar and they have the same eigenvalues. Since $\mathbf{S}$ is symmetric, it is diagonalizable with a set of M eigenvalues

$$\lambda_0 \geqslant \lambda_1 \geqslant \cdots \geqslant \lambda_{M-1}, \tag{4.9}$$

whose eigenvectors $\mathbf{U}_j$, $j = 1, \ldots, M$ form an orthonormal basis of $\mathbb{R}^M$. The right eigenvectors of $\mathbf{K}$ are given by

$$\mathbf{V}_j = \mathbf{D}^{-1/2}\mathbf{U}_j. \tag{4.10}$$

Since $\mathbf{K}$ is a Markov matrix, all its eigenvalues are smaller than or equal to 1 in absolute value. Moreover, if the parameter σ in Eq. (4.1) is large enough [and, thus, the norm vector in Eq. (4.4) is "small enough"], all points are (numerically) connected and the largest eigenvalue $\lambda_0 = 1$ has multiplicity 1 with corresponding right eigenvector

$$\mathbf{V}_0 = [1,1,\ldots,1]. \tag{4.11}$$

We define the n-dimensional representation of N-dimensional state vectors by the following *diffusion map*:

$$\Psi_n: \mathbf{X}^{(i)} \to [V_1^{(i)}, V_2^{(i)}, \ldots, V_n^{(i)}]; \tag{4.12}$$

that is, the point $\mathbf{X}^{(i)}$ is mapped to a vector containing the ith coordinate of each of the first n leading eigenvectors of the matrix $\mathbf{K}$. This mapping Ψ_n: $\mathbb{R}^N \to \mathbb{R}^n$ is defined *only* at the M recorded state vectors. We will show later that it can be extended to nearby points in the N-dimensional phase space, without full recomputation of a new matrix and its eigenvectors. In Appendix A we provide a theoretical justification for this method as a dynamically useful dimensionality reduction step.

B. Computation of data-based observables

We replaced Eq. (4.1) by Eq. (4.4) where the weight vector (4.2) needs to be further specified. Two natural choices for the values of components of the weight vector $\mathbf{a}=[a_1,a_2,\ldots,a_N]$ immediately arise. One option is to regard the absolute values of the components of the state vector $\mathbf{X}$ as of "equal importance," i.e.,

$$a_k = \omega \quad \text{for } k = 1,2,\ldots,N, \tag{4.13}$$

where ω is a single method parameter; this is identical to the use of a single σ in Eq. (4.1), namely $\sigma=\omega^{-1}$.

The above approach uses the Euclidean distance between data vectors as the basis for graph Laplacian construction and eigenanalysis. In our case, the components of these vectors are concentrations of different species (e.g., integer numbers of protein molecules, each with its own range over the data set). Moreover, the data vectors contain integers (0 and 1) representing states of Boolean operators. This motivates a second natural choice of the weight vector $\mathbf{a}=[a_1,a_2,\ldots,a_N]$. We rescale the state vector X to span the symmetrical domain (cube) in N-dimensional space, i.e.,

$$a_k = \frac{\tilde{\omega}}{\max_i X_k^{(i)} - \min_i X_k^{(i)}} \quad \text{for } k = 1,2,\ldots,N, \tag{4.14}$$

where the maximum and minimum values are computed over all $i=1,\ldots,M$. Formula (4.14) implies that components of the vector $\mathbf{X}^{(i)}-\mathbf{X}^{(j)}$, $i,j=1,\ldots,M$, satisfy

$$X_k^{(i)} - X_k^{(j)} \in [-\tilde{\omega},\tilde{\omega}] \quad \text{for } k = 1,\ldots,N, \text{ and } i,j = 1,\ldots,M.$$

The difference between Eqs. (4.13) and (4.14) is that the first formula implicitly assumes that the fluctuations in different components of the state vector $\mathbf{X}$ are equally important, i.e., the absolute values of fluctuations are important. Formula (4.14) on the other hand implies that *relative* changes (compared to the maximal observed change) in each component are more representative than the absolute values of the changes. We will see below that Eq. (4.13) appears more suitable for our variable-free analysis.

TABLE I. Top eigenvalues of matrix $\mathbf{K}$ computed using Eq. (4.13) for $\alpha=0$ in Eq. (4.5).

ω	λ_0	λ_1	λ_2	λ_3
0.02	1.000 00	0.999 86	0.945 06	0.913 60
0.01	1.000 00	0.999 20	0.777 57	0.711 22
0.005	1.000 00	0.992 79	0.443 52	0.355 15
0.002	1.000 00	0.762 62	0.107 15	3.3×10^{-2}
0.001	1.000 00	0.283 46	1.2×10^{-2}	1.1×10^{-3}
0.0005	1.000 00	7.5×10^{-2}	1.0×10^{-3}	1.5×10^{-4}

1. Comparison of Formulae (4.13) and (4.14)

Using our illustrative gene regulatory network example (2.1)–(2.3) we now study the dependence of the eigenvectors of the matrix $\mathbf{K}$ on the weighting vector $[a_1,a_2,\ldots,a_N]$. We run the long-time Gillespie based stochastic simulation of Eqs. (2.1)–(2.3) to obtain a representative set of M state vectors using the following dimensionless stochastic rate constants $\gamma_1=\gamma_2=1.14$, $\varepsilon_1=\varepsilon_2=0$, $\delta_1=\delta_2=7.5\times10^{-4}$, $k_1=k_2=10^{-3}$, $k_{-1}=k_{-2}=10$, $k_{o1}=k_{o2}=0.4$, $k_{-o1}=k_{-o2}=10$. After removing initial transients, we started recording the values of the state vector (2.4) every 2×10^8 SSA time steps. We made 2000 recordings to obtain a data file with $M=2000$ state vectors. Next, we use these state vectors $\mathbf{X}^{(i)}$ to compute the $M\times M$ matrix $\mathbf{K}$ and its eigenvectors. We use formula (4.13) to compute $\mathbf{W}$ and $\mathbf{D}$ by Eqs. (4.4)–(4.6). Then we use implicitly restarted Arnoldi methods (ARPACK package[28]) to find the eigenvectors corresponding to the highest eigenvalues of the symmetric matrix $\mathbf{S}$ given by Eq. (4.8). Finally, we compute the eigenvectors of $\mathbf{K}=\mathbf{D}^{-1}\mathbf{W}$ by Eq. (4.10).

The formula (4.13) has a single parameter ω which is free for us to specify. It is easy to check numerically that the larger the "local neighborhood" size selected (that is, the smaller the ω value) the denser the connections between data points in the graph. Table I shows the highest eigenvalues for different values of ω. We already know from Ref. 4 that the system is effectively one dimensional. A good observable for the system is known to be $Q=P_2-P_1$, i.e., the difference between the first two coordinates of the state vector. However, the protein concentrations P_1 or P_2 were also found to give good equation-free results.

We plot the "empirical" good observable of each data point i [its P_1 component, i.e., $X_1^{(i)}$, or the difference of its P_1 and P_2 components, i.e., $Q=X_2^{(i)}-X_1^{(i)}$] versus the one-dimensional representation $\Psi_1(\mathbf{X}^{(i)})$ [see Eq. (4.12)] of the point. The results are given in Fig. 2 for two different values of ω. The fact that the empirical coordinate Q appears to effectively be *one to one* with the "automated" coordinate $\Psi_1(\mathbf{X}^{(i)})$ for all points in the data set confirms that Q is indeed a good coordinate for data representation [the figure clearly shows Q as the graph of a function above $\Psi_1(\mathbf{X}^{(i)})$, i.e., that the relation between Q and $\Psi_1(\mathbf{X}^{(i)})$ is one to one]. The P_1 vs $\Psi_1(\mathbf{X}^{(i)})$ graph confirms that P_1 is also a good observable; it is also approximately one to one with $\Psi_1(\mathbf{X}^{(i)})$, yet the slightly "fat curve" suggests that Q is a "better" observable.

The dependence of the variable-free results on the value chosen for ω may be rationalized through Eq. (4.1). As dis-

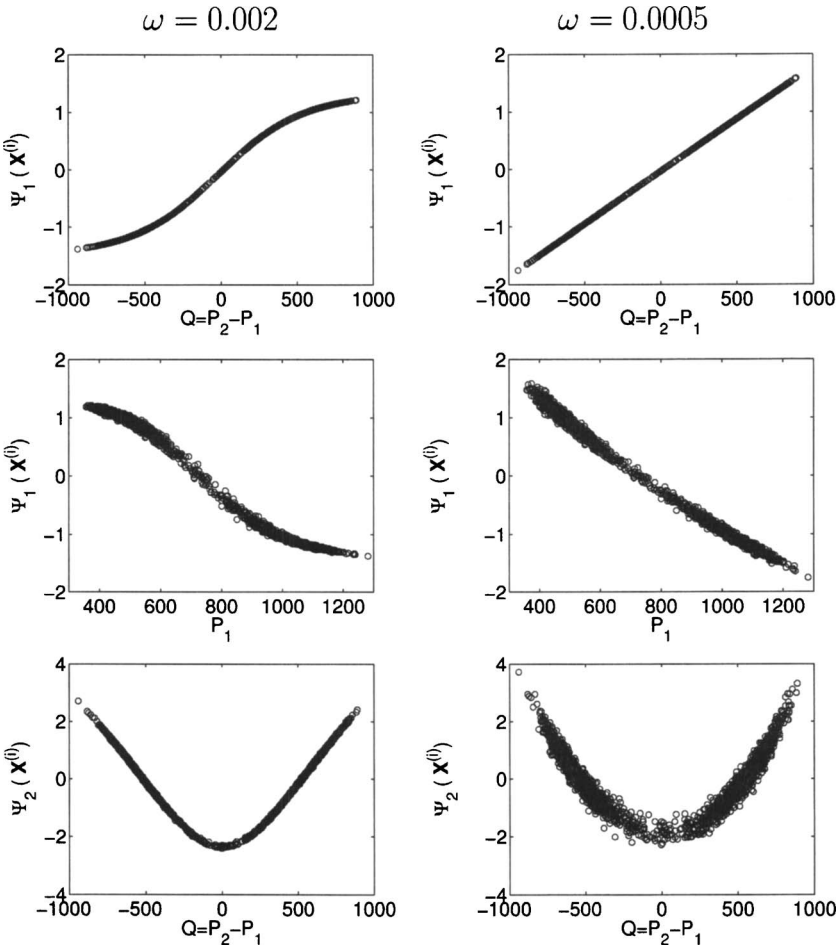

FIG. 2. Variable-free results using formula (4.13) and $\omega=0.002$ (left panels) or $\omega=0.0005$ (right panels). We plot $\Psi_1(\mathbf{X}^{(i)})$ which corresponds to eigenvalue λ_1 as a function of $Q=P_2-P_1$ (top panels) and as a function of P_1 (center panels). We also plot $\Psi_2(\mathbf{X}^{(i)})$ which corresponds to eigenvalue λ_2 as a function of $Q=P_2-P_1$ (bottom panels).

cussed in Sec. IV B, our parameter ω is analogous to an inverse "cutoff length" in the computation of the diffusion map kernel; if it is too large, then the graph becomes disconnected. Clearly, it is a model parameter that has to be optimized depending on the problem; our results for $\omega=0.0005$ show a pure linear relation between the "empirical" Q and the "automated" $\Psi_1(\mathbf{X}^{(i)})$ observables. Increasing ω by a factor of 2 corresponds to raising the elements of the matrix $\tilde{\mathbf{W}}$ to the fourth power. This change in weight factor [followed by the normalization of Eq. (4.6)] leads to a different clustering of the data points. Large ω implies that Euclidean distances are meaningful when small; this results in a "more clustered" data set, where nearby data points (e.g., points within one potential well) appear (in diffusion map coordinates) relatively closer, while points far away (e.g., points in different potential wells) appear (in diffusion map coordinates) relatively more distant. Indeed, in the case of continuous variables, in the limit of large ω the eigenvectors of the diffusion map converge to the eigenfunctions of a corresponding Fokker-Planck diffusion operator. In the case of two deep potential wells, this eigenfunction is approximately constant in the two wells with a sharp transition between them. This might explain the slightly flat regions at the two edges of the apparent curve in the middle panel of Fig. 2 for $\omega=0.002$; points within the same potential well may differ in Q, yet appear more nearby in the "automated" observable. We also include a plot of the relation between Q and the component of the data in the second eigenvector $\Psi_2(\mathbf{X}^{(i)})$ for comparison.

Next we show that the weight vector computed using the formula (4.14) (based on the magnitude of *relative* state variable changes) is unsuitable for our variable-free analysis. We use the same set of $M=2000$ state vectors $\mathbf{X}^{(i)}$ to compute the $M\times M$ matrix $\mathbf{K}$ and its eigenvectors, using formula (4.14) to compute $\mathbf{W}$ and $\mathbf{D}$ by Eqs. (4.4)–(4.6). A single parameter $\tilde{\omega}$ still remains to be specified in formula (4.14). We now again compare the "empirical" and "automated" observables of all data points [$Q=P_2-P_1$ as a function of $\Psi_1(\mathbf{X}^{(i)})$, the one-dimensional representation based on the first nontrivial eigenvector of the matrix $\mathbf{K}$]. The results are given in Fig. 3 for two different values of $\tilde{\omega}$. We see that the data split into four curves. Each curve corresponds to a distinct combination of gene operator states (actually, two of the curves effectively coincide). There are exactly four possibilities of gene states taken from the set

$$[O_1,O_2] \in \{[0,0],[0,1],[1,0],[1,1]\}.$$

If we use formula (4.13), then the contribution of the distance between gene operator states to the data Euclidean distance is negligible compared to the fluctuations of the protein numbers. Local distances computed using the scaling in formula (4.14) are clearly *not* representative of the similarity of nearby (in this metric) points for the system dynamics: there is no one-to-one correspondence between the empirically known "good observable" Q and the "automated" $\Psi_1(\mathbf{X}^{(i)})$. Indeed, for the parameter values of our simulation, transitions between the 0 and 1 states of the operators are *very* fast ("easy"); on the other hand the Euclidean distance of two

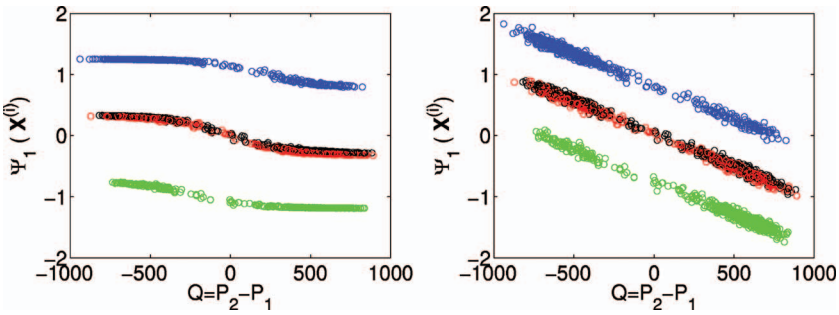

FIG. 3. (Color) Variable-free results using formula (4.14) and $\tilde{\omega}=2$ (left panel) or $\tilde{\omega}=0.1$ (right panel); data points colored according to gene states: black=[0,0], green=[0,1], blue=[1,0], and red=[1,1]. We plot $\Psi_1(\mathbf{X}^{(i)})$ which corresponds to eigenvalue λ_1 as a function of $Q=P_2-P_1$.

data points that differ only in these states is large when computed through the formula (4.14).

An alternative approach to computing the effective rate in Eq. (2.1) can be obtained assuming that reaction (2.2) is fast and that we have a lot of protein molecules in the system. Then the quasi-steady-state assumption gives the formula $\overline{P_1P_1}=2k_1/k_{-1}P_1^2$. Hence, we can write the number of dimers as a simple function of the number of monomer proteins. On the other hand, using the same approximation in Eq. (2.3), we obtain

$$O_1=\frac{k_{-o1}}{k_{o1}\overline{P_2P_2}+k_{-o1}}. \tag{4.15}$$

Equation (4.15) gives O_1 as a real number in the interval [0, 1]. This number is a good approximation for computing the effective rate in Eq. (2.1). However, it is not a value *of the Boolean variable* O_1. It is only a probability that the gene "is on" at the given time.

If, on the other hand, the "on-off" operator transitions were slow, then Fig. 3 would be quite informative: it would suggest that we should *augment* our observables with the Boolean variables O_1 and O_2, since these are "slow." Because of the Boolean nature of the gene operator variables, it is not possible to know *a priori* how often these transitions occur, and, consequently, how to scale the quantized Boolean state distance so that it "meaningfully" participates in the Euclidean distance used for diffusion map analysis. As our diffusion map computations stand, we do not take into account the *temporal* proximity of points—when they have been obtained from the same transient. If such information is taken into account, it is conceivable that temporal proximity would provide guidance in choosing the components of weight vectors (especially for Boolean variables which change in a quantized manner) so that "local" Euclidean distances are indeed representative of the dynamical proximity between data points.

V. VARIABLE-FREE COMPUTATIONS

We now couple the above automated detection of observables with the equation-free computations in Ref. 4 in what we will refer to as "variable-free, equation-free" methods. The results in this section are for the model parameter values given in Sec. IV B 1 using the weight vector defined by Eq. (4.13) with $\omega=0.0005$ and kernel parameter $\alpha=0$ (the standard, normalized graph Laplacian) in Eq. (4.5).

The data plot in terms of the observable Q and the component in the eigenvector $\Psi_1(\mathbf{X}^{(i)})$ in Fig. 2 suggested that a single diffusion map coordinate, denoted $Q_{\text{dmap}}\equiv\Psi_1(\mathbf{X}^{(i)})$, is sufficient to characterize the system dynamics. The diffusion map coordinate is found by performing the eigencomputations described in Sec. IV A using the full state vector ($N=6$) at each of the $M=2000$ recorded SSA data points (every 2×10^8 SSA time steps) as input to our numerical routines.

In our previous paper[4] we described an approach to compute an effective free energy potential in terms of the observable $Q=P_2-P_1$. Variable-free computation of the effective free energy is now feasible using a similar approach modified to analyze simulation data in terms of the coordinate Q_{dmap}. Figure 4 plots the effective potential $\beta\Phi$ in terms of the automated reduction coordinate Q_{dmap}. To evaluate the effective drift (V) and diffusion (D) coefficients required in the construction of the effective free energy [Eq. (3.4)] we choose a value of Q_{dmap}, locate instances when it appears in the simulation database, record its subsequent evolution within a fixed time interval, and then average over these instances to estimate the rate of change in the mean and the variance. This procedure is repeated for a grid of Q_{dmap} values enabling numerical evaluation of the integral in Eq. (3.4). The result of this analysis is compared in Fig. 4 with the potential obtained by directly constructing the probability distribution $f(q_{\text{dmap}})$ from the time series and employing the relationship $\beta\Phi(Q_{\text{dmap}})\sim-\log[f(q_{\text{dmap}})]$.

Section V B describes a *lifting* procedure that allows short bursts of simulation, instead of long-time simulation, to be used in variable-free estimation of effective drift and diffusion coefficients. The central idea of "variable-free,

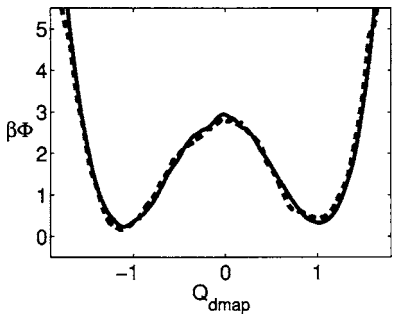

FIG. 4. Effective free energy $\beta\Phi$ as a function of Q_{dmap} from binning of all data points using a SSA database of 2^{37} time steps (solid line) and computed from numerical integration of Eq. (3.4) using a 2^{34} point subsampling (keeping 1 out of every 8 points) of this database (dashed line). Numerical integration performed using a more severe subsampling of the database with 2^{31} points produces an effective free energy profile with an unacceptable level of noise.

equation-free" methods is to perform equation-free analysis *in terms of diffusion map variables*, based on short bursts of SSA simulation *in the original variables*. This strategy requires an efficient means of converting between the physical variables of the system and those of its diffusion map (a *restriction* step) and vice versa: *lifting* from the diffusion map back to physical variables. For small sample sizes, eigendecomposition of the symmetric kernel $\mathbf{S}$ [defined in Eq. (4.8)] yields the diffusion map variables *for each data point*; yet, as the number of sample data points increases, the associated computational costs become prohibitive. The Nyström formula[29,30] for eigenspace interpolation is a viable alternative to repeated matrix eigendecompositions for computing diffusion map coordinates of new data points generated during the course of a simulation. Eigenvectors and eigenvalues of the kernel $\mathbf{S}$ are related by $\mathbf{S}\mathbf{U}_j=\lambda_j\mathbf{U}_j$, or equivalently

$$U_j(\mathbf{X}^{(i)}) = \frac{1}{\lambda_j}\sum_{k=1}^{M} S_{ik}U_j(\mathbf{X}^{(k)}), \tag{5.1}$$

where $U_j(\mathbf{X}^{(i)})$ denotes the component of the jth eigenvector associated with state vector $\mathbf{X}^{(i)}$. Eigenvector components associated with a *new* state vector $\mathbf{X}^{\text{new}}$ cannot be computed directly from Eq. (5.1) because entries of the matrix $\mathbf{S}$ are defined only between pairs of data points in the original data set. Defining the $M\times 1$ vector $\hat{\mathbf{W}}^{\text{new}}$ of exponentials of the negative squares of the distances between the new point and database points by

$$\hat{W}_i^{\text{new}} = \exp[-\|\mathbf{X}^{\text{new}} - \mathbf{X}^{(i)}\|_{\mathbf{a}}^2], \tag{5.2}$$

and the $M\times 1$ vector $\mathbf{W}^{\text{new}}$ by

$$W_i^{\text{new}} = \left(\sum_{k=1}^{M}\tilde{W}_{ik}\right)^{-\alpha}\left(\sum_{k=1}^{M}\hat{W}_k^{\text{new}}\right)^{-\alpha}\hat{W}_i^{\text{new}} \tag{5.3}$$

allows the generalized kernel vector $\mathbf{S}^{\text{new}}$ to be defined as follows:

$$S_i^{\text{new}} = \left(\sum_{k=1}^{M}W_{ik}\right)^{-1/2}\left(\sum_{k=1}^{M}W_k^{\text{new}}\right)^{-1/2}W_i^{\text{new}}. \tag{5.4}$$

The entries in $\mathbf{S}^{\text{new}}$ quantify the pairwise similarities between the new point $\mathbf{X}^{\text{new}}$ and database points consistent with the definition of $\mathbf{S}$ in Eq. (4.8).[30]

A. *Restriction* from physical to diffusion map variables

The Nyström formula[29] is used to find the eigenvector component $U_j(\mathbf{X}^{\text{new}})$ associated with a new state vector $\mathbf{X}^{\text{new}}$

$$U_j(\mathbf{X}^{\text{new}}) = \frac{1}{\lambda_j}\sum_{i=1}^{M} S_i^{\text{new}}U_j(\mathbf{X}^{(i)}), \tag{5.5}$$

allowing the eigenvectors of the matrix $\mathbf{K}$ (and thereby the diffusion map coordinates) associated with $\mathbf{X}^{\text{new}}$ to be computed using Eq. (4.10). A full eigendecomposition is typically performed first for a representative subset of the (large) number of SSA data points and the Nyström formula is then used to perform the restriction operation in Eq. (5.5) which amounts to interpolation in the diffusion map space.

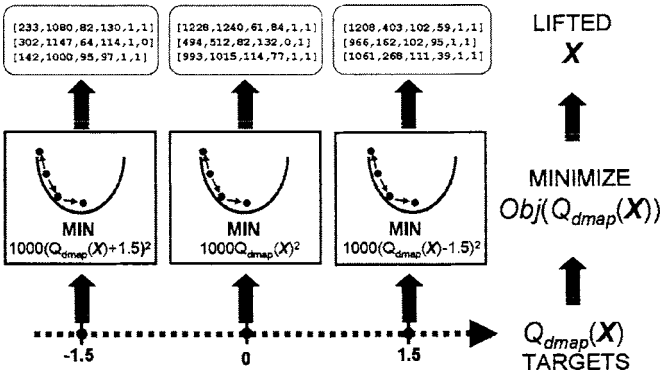

FIG. 5. A schematic of the procedure for lifting from diffusion map coordinate $Q_{\text{dmap}}(\mathbf{X})$ to six-dimensional state vector $\mathbf{X}$ via minimization of quadratic constraint potential $\text{Obj}(Q_{\text{dmap}}(\mathbf{X}))$. Target values of the diffusion map coordinate are shown at the base of the figure, with the potential function to be minimized in each case indicated above these targets. For each diffusion map coordinate value shown, three consistent state vectors (generated by lifting) are indicated at the top of figure.

B. *Lifting* from diffusion map to physical variables

The process of *lifting* (shown schematically in Fig. 5) consists of preparing a detailed state vector with prescribed diffusion map coordinates $Q_{\text{dmap}}^{\text{targ}}$. The main step in our lifting process is the minimization of the following objective function:

$$\text{Obj}(Q_{\text{dmap}}(\mathbf{X})) = \lambda_{\text{obj}}(Q_{\text{dmap}}(\mathbf{X}) - Q_{\text{dmap}}^{\text{targ}})^2, \tag{5.6}$$

where λ_{obj} is a weighting parameter that controls the shape of the objective away from its minimum at $Q_{\text{dmap}}(\mathbf{X}^*)=Q_{\text{dmap}}^{\text{targ}}$. The objective function is a smooth function of $Q_{\text{dmap}}(\mathbf{X})$ but is not necessarily as smooth in the physical variables $\mathbf{X}$. The implicit dependence of Q_{dmap} on $\mathbf{X}$ makes this optimization problem nontrivial.

We use here, for simplicity, the method of simulated annealing[31,32] (SA) to solve the optimization problem, and identify a value of the state vector $\mathbf{X}^*$ with the target diffusion map coordinates $Q_{\text{dmap}}^{\text{targ}}$. SA is attractive since it does not require calculation of derivatives of $Q_{\text{dmap}}(\mathbf{X})$ with respect to the physical variables. The SA routine[32] employs a "thermalized" downhill simplex method as the generator of changes in configuration. The simplex, consisting of $N+1$ vertices, each corresponding to a trial state vector, tumbles over the objective landscape defined by Eq. (5.6) sampling new state vectors as it does so. The control parameter of the method is the "annealing temperature" which controls the rate of simplex motion. At high temperatures the method behaves like a global optimizer, accepting many proposed configurations (even those that take the simplex uphill, i.e., in the direction of increasing objective function value). At low temperatures a local search is executed and only downhill simplex moves are accepted. The objective defined in Eq. (5.6) has numerous local minima in $\mathbf{X}$ [i.e., many different vectors of physical variables $\mathbf{X}$ satisfy $Q_{\text{dmap}}(\mathbf{X})=Q_{\text{dmap}}^{\text{targ}}$] and, for our purposes, it is sufficient to locate any such local minimum; a modest, computationally inexpensive, annealing schedule suffices for this.

The starting simplex configuration for this N-parameter minimization may be selected at random or (more reason-

ably) by taking those state vectors in the existing database with diffusion map coordinates closest to the target $Q_{\text{dmap}}^{\text{targ}}$. It is important to note that the SA optimization scheme *requires* the Nyström formula at each iteration to compute $Q_{\text{dmap}}(\mathbf{X}^{\text{trial}})$ for trial state vectors, and thus evaluate the objective function value, which determines whether the configuration will be accepted or not. Once the objective has been evaluated at each of the starting vertices, the following steps are repeated until a minimum is located:

(a) move the simplex to generate a new state vector $\mathbf{X}^{\text{trial}}$;
(b) evaluate the objective function value at the new state vector $\text{Obj}(Q_{\text{dmap}}(\mathbf{X}^{\text{trial}}))$; and
(c) decrement the annealing temperature.

The downhill simplex method prescribes the motion in step (a) making a selection from a set of moves according to the local objective "terrain" (set of objective values at the vertices encountered). Step (b) requires an evaluation using the Nyström formula. We note here that this lifting strategy prepares state vectors with desired diffusion map coordinates using search algorithm "dynamics". The suitability of this approach relative to alternatives that employ physical dynamics (e.g., using constrained evolution of the stochastic simulator in the spirit of the SHAKE algorithm in molecular dynamics[33]) is a relevant and interesting question that merits further investigation.

C. Illustrative numerical results

Equipped with restriction and lifting operators between physical and "automated" variables, we can now perform all the equation-free tasks of Ref. 4 in the diffusion map coordinate Q_{dmap}, i.e., in variable-free mode.

A procedure for variable-free computational estimation of $V(q)$ and $D(q)$ in Eq. (3.4) is as follows:

(A) At the value $Q_{\text{dmap}}=q_{\text{dmap}}$ lift to a consistent state vector using the approach described in Sec. V B.
(B) Use the state vector computed in step (A) as an initial condition for a short simulation burst and run multiple realizations for time Δt. Restrict the results of these simulations (Sec. V A) and use definitions (3.2) and (3.3) [with $Q_{\text{dmap}}(t)$ instead of $Q(t)$] to estimate the effective drift $V(q_{\text{dmap}})$ and the effective diffusion coefficient $D(q_{\text{dmap}})$.
(C) Repeat steps (A) and (B) for sufficiently many values of Q_{dmap} and then compute $\Phi(q)$ using formula (3.4) and numerical quadrature.

We performed lifting for three values of the automated reduction coordinate $(Q_{\text{dmap}}=-1.5,0,1.5)$, generating several replicas in each case. From Fig. 6 it is apparent that the selected values of Q_{dmap} are located near the "rims" of the wells of two local minima on the effective free energy landscape for this system. The state vectors generated by lifting are shown at the top of Fig. 5. Figure 6 plots the SSA simulation evolution, initialized at these state vectors, in the observable Q_{dmap}. Also shown in Fig. 6 is the steady-state distribution in terms of Q_{dmap} obtained from long SSA runs. Estimates for drift (V) and diffusion (D) coefficients at Q_{dmap} values of -1.5 and 0 produced by sampling the simulation database and using the lifting procedure described in this paper are compared in Table II. It should be possible to reach a better agreement between the coefficient estimates based on the long simulation database and those obtained by a lifting procedure if we evolve the actual model dynamics with a constraint on the prescribed Q_{dmap} value—possibly through a parabolic constraint potential of the type used in umbrella sampling (see also the "run and reset" procedure described in Refs. 4 and 34). The effective free energy predicted by analyzing the full simulation database in terms of Q_{dmap} can be found in Fig. 4.

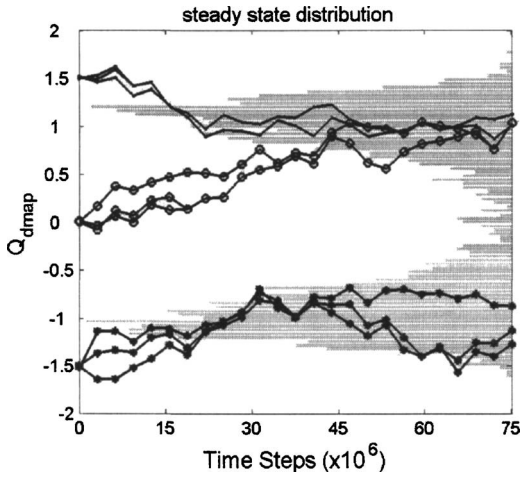

FIG. 6. Drift in the diffusion map coordinates. The shaded horizontal boxes indicate the steady-state probability distribution for $M=2000$. Points from SSA trajectories are shown at intervals of 3×10^6. Initial configurations for these runs are those shown in Fig. 5 prepared by lifting from Q_{dmap} values of $(-1.5,0,1.5)$. Trajectories drift towards the most populated regions of the distribution.

VI. SUMMARY AND CONCLUSIONS

The knowledge of good observables is vital in our ability to create effective reduced models of complex systems, and thus to analyze and even design their behavior at a macroscopic/engineering level more efficiently. In this paper we illustrated a connection between computational data mining (in particular, diffusion maps and the resulting low-dimensional description of high-dimensional data) with computational multiscale methods (in particular, certain equation-free algorithms). Our illustrative example consisted of a model gene regulatory network known to exhibit bistable (switching) behavior in some regime of its parameter space. We also presented examples of *lifting* and *restriction* protocols that enable the passing of information between

TABLE II. Estimates for drift (V) and diffusion (D) coefficients at Q_{dmap} values of -1.5 and 0 using initial conditions drawn from the simulation database and prepared by lifting.

	Database		Lifting	
$(Q_{\text{dmap}})_0$	V	D	V	D
-1.5	3.3×10^{-5}	4.7×10^{-6}	2.1×10^{-5}	3.2×10^{-6}
0	5.3×10^{-6}	4.0×10^{-6}	7.9×10^{-8}	4.1×10^{-6}

detailed state space and reduced "diffusion map coordinate" space. These protocols allow us to "intelligently" design short bursts of appropriately initialized stochastic simulations with the detailed model simulator. Processing the results of these simulations *in diffusion coordinate space* forms the basis for the design of subsequent numerical experiments aimed at elucidating long-term system dynamic features (such as equilibrium densities, effective free energy surfaces, escape times between different wells, and their parametric dependence). In particular, we confirmed that previously, empirically known, observables were indeed meaningful coarse-grained coordinates.

In traditional diffusion map computations, a single scalar (a scaled Euclidean norm) forms the basis for the identification of good reduced coordinates (when they exist). An important issue that arose in our example, due to the disparate nature, value ranges, and dynamics of different data vector components, was the selection of appropriate *relative* scaling among data component values. The computational approach we used was based on the data ensemble, without any contribution from the *dynamical proximity* between data points collected along the same trajectory. We believe that incorporating such information will be very useful in determining relative scalings among disparate data components; finding ways to integrate such information among data ensembles collected in different experiments, and possibly with different sampling rates will greatly assist in this direction.

Our illustrative example consists of a simple caricature of the genetic switch described by the six-dimensional state vector $\mathbf{X}$ given by Eq. (2.4). More realistic gene regulatory networks are described by the state vector $\mathbf{X}$ given by Eq. (1.1) whose dimension N ranges from tens to hundreds of species. It is worth noting that large values of N do not complicate the variable-free part of the algorithm. The dimensionality N of the microscopic model appears only in the computation of the weighted Euclidean norm in Eq. (4.4). $\tilde{\mathbf{W}}$ computed in Eq. (4.4) is an $M \times M$ matrix where M is the number of points in the data set considered. The computational intensity of the diffusion map part of the algorithm scales therefore with M and is independent of the dimensionality N of the original data set. Of course, the stochastic simulation from which the data for the diffusion map approach are collected depends on both the number of species and the nature (e.g., stiffness) of the model dynamics. After the data collection process, the diffusion map computations presented will, in principle, require the same effort for the same number of data points and systems that have the same *effective* dimension n. It is precisely this *effective* dimensionality n of the reduced (macroscopic) problem which determines the applicability of the equation-free methods.

In this work, diffusion map computations were based on data collected from a single long transient that was considered representative of the entire relevant portion of the (six-dimensional) phase space. In more realistic problems such long simulations will be no longer possible; yet local simulation bursts, observed on *locally valid* diffusion map coordinates, can be used to guide the efficient exploration of phase space. Local smoothness in these coordinates allows us to use them in protocols such as umbrella sampling[33,35] to "differentially locally extend" effective free energy surfaces. For example, "reverse coarse" integration described in Refs. 36 and 37 provides computational protocols for microscopic/stochastic simulators to track backward in time behavior, accelerating escape from free energy minima and allowing identification of saddle-type coarse-grained "transition states". Design of (computational) experiments for obtaining macroscopic information is thus complemented by the design of (computational) experiments to extend good low-dimensional data representations: both the coarse-grained coordinates *and* the operations we perform on them can be obtained through appropriately designed fine scale simulation bursts.

In this paper the connection between diffusion maps and coarse-grained computation operated only in one direction: diffusion map coordinates influenced the subsequent design of numerical experiments. An important current research goal is to establish the "reverse connection:" the on-line extension/modification of diffusion map coordinates towards sampling important, unexplored regions of phase space.

ACKNOWLEDGMENTS

This work was partially supported by DARPA [for four of the authors (T.A.F., R.C., I.G.K., and B.N.)], the Israel Science Foundation Grant No. 432/06 [to one of the authors (B.N.)], NIH Grant No. R01GM079271-01 [to two of the authors (T.C.E. and X.W.)], and the Biotechnology and Biological Sciences Research Council Grant No. BB/C508618/1 and Linacre College, University of Oxford [to one of the authors (R.E.)].

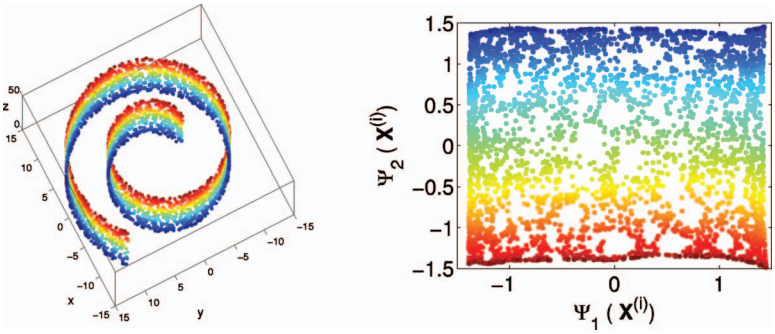

FIG. 7. (Color) Left panel: Swiss roll data set in $\mathbb{R}^3$. Data points lie along a two-dimensional manifold. Data points are colored by their z-coordinate value (ordering of data points passed to diffusion map routine is random). Right panel: plot of $\Psi_1(\mathbf{X}^{(i)})$ (corresponding to eigenvalue λ_1) against $\Psi_2(\mathbf{X}^{(i)})$ (corresponding to eigenvalue λ_2) for points in the data set (same coloring scheme). The diffusion map "unrolls" the two-dimensional manifold.

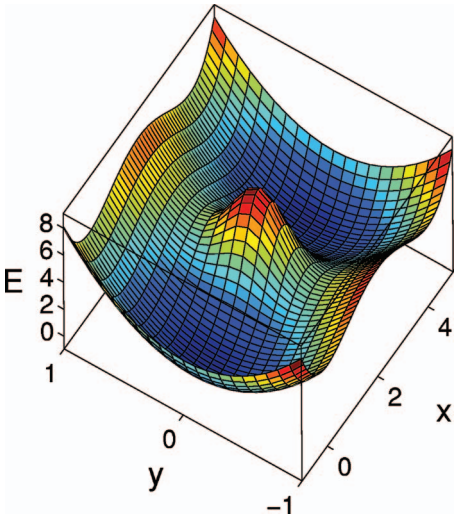

FIG. 8. (Color) Two-well potential with two connecting pathways between minima.

APPENDIX A: DIFFUSION MAPS

The following discussion is largely adapted from Refs. 11 and 27. We present a criterion for dimensionality reduction and show how it leads to the diffusion map method.

Suppose we have M points $\mathbf{X}^{(i)} \in \mathbb{R}^N$, $i=1,\ldots,M$, and we define the matrix $\mathbf{W}$ by Eq. (4.5). Given a mapping $\mathbf{f}:[1,\ldots,M]\to\mathbb{R}^n$, we define the functional $\mathcal{L}$ by the formula

$$\mathcal{L}(\mathbf{f}) = \sum_{i,j} \|\mathbf{f}(i) - \mathbf{f}(j)\|^2 W_{ij}. \tag{A1}$$

We see that $\mathcal{L}(\mathbf{f})$ is always non-negative. Moreover, W_{ij} is close to (far from) 1 for vectors $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ which are near (far) from each other. For a dimensionality reduction function $\mathbf{f}$ to be useful, we must make sure that nearby points $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ in $\mathbb{R}^N$ are mapped to nearby points $\mathbf{f}(i)$ and $\mathbf{f}(j)$ in $\mathbb{R}^n$. To find such a mapping, one can solve the following minimization problem:

$$\arg\min_{\mathbf{f}\in\mathbb{F}} \mathcal{L}(\mathbf{f}) \quad \text{where } \mathbb{F} = \{\mathbf{f}:\mathbf{F}^T\mathbf{D}\mathbf{F} = \mathbf{I}_n, \mathbf{F}^T\mathbf{D}\mathbf{1} = \mathbf{0}\}, \tag{A2}$$

where $\mathbf{F}$ is the $M\times n$ matrix with row vectors $\mathbf{f}(i)$, $\mathbf{D}$ is the $M\times M$ diagonal matrix with entries $D_{ii}=\Sigma_j W_{ij}$, $i=1,\ldots,M$, $\mathbf{I}_n$ is the $n\times n$ identity matrix, $\mathbf{1}$ is a vector of M ones, and $\mathbf{0}$ a vector of n zeros. The first constraint removes the arbitrary scaling factor, while the second constraint ensures that we do not map all M points $\mathbf{X}^{(i)}$ to the same number. Since Eq. (A1) can be rewritten as

$$\mathcal{L}(\mathbf{f}) = \sum_{i,j=1}^{M} \|\mathbf{f}(i) - \mathbf{f}(j)\|^2 W_{ij} = \mathrm{tr}(\mathbf{F}^T(\mathbf{D}-\mathbf{W})\mathbf{F}), \tag{A3}$$

the solution $\mathbf{F}$ is given by the matrix of eigenvectors corresponding to the lowest eigenvalues of the matrix

$$\mathbf{D}^{-1}[\mathbf{D}-\mathbf{W}] = \mathbf{I}_M - \mathbf{K} \tag{A4}$$

or equivalently by the largest eigenvalues of $\mathbf{K}$. By the non-negativity of the functional $\mathcal{L}(\mathbf{f})$ it follows that the eigenvalues of $\mathbf{I}_M-\mathbf{K}$ are all non-negative, or that all eigenvalues of $\mathbf{K}$ are smaller than or equal to 1. The eigenvector corresponding to the eigenvalue $\lambda_0=1$ is the vector $\mathbf{1}$. Ordering the remaining eigenvalues in decreasing order we see that the n-dimensional representation of N-dimensional data points, via the minimization of Eq. (A2), is the diffusion map (4.12).

We note that our M points and the matrix W_{ij} can be also viewed as a weighted full graph with M vertices, where the weight associated with an edge between points i and j is equal to W_{ij}. Then the previous analysis can be reformulated in terms of standard spectral graph theory.[27,38] More precisely, it was shown in Ref. 18 that this construction leads to the classical normalized graph Laplacian for $\alpha=0$ in Eq. (4.5). If $\alpha=1$, then the construction gives the *Laplace-Beltrami* operator on the graph. Finally, if the data are produced by a stochastic (Langevin) equation, $\alpha=1/2$ provides a consistent method to approximate the eigenvalues and eigenvectors of the underlying stochastic problem.

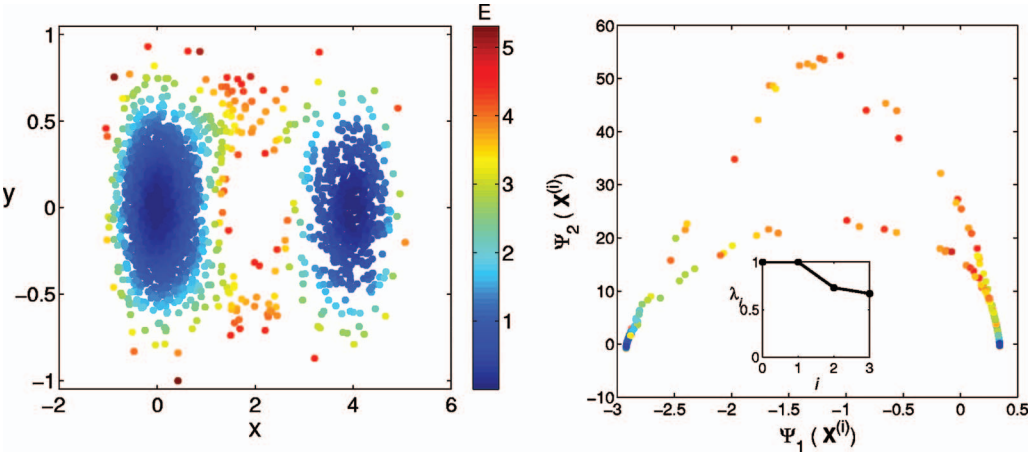

FIG. 9. (Color) Left panel: subsampled data set generated by Monte Carlo simulation using the two-well potential (data points colored by energy according to color bar). Right panel: data set diffusion map (same coloring scheme) with top eigenvalues indicated in inset.

APPENDIX B: SIMPLE ILLUSTRATIVE EXAMPLES

We include a brief illustration of the application of the diffusion map approach to the well known three-dimensional "Swiss roll" data set[39–41] (shown in the left panel of Fig. 7) where data points lie along a two-dimensional manifold. For this data set $\mathbf{X}=[x,y,z]$; to compute the diffusion map we use $\alpha=1$, and $\sigma=2$ in Eq. (4.1). Figure 7 (right panel) plots these data points in terms of their components in the top two significant eigenvectors $[\Psi_1(\mathbf{X}^{(i)})$ and $\Psi_2(\mathbf{X}^{(i)})]$ of the matrix $\mathbf{K}$ for this data set; it shows the "unrolled" two-dimensional manifold detected by the diffusion map algorithm. The same result is obtained irrespective of the ordering (or orientation) of the data set used to compute the pairwise similarity matrix.

As a second illustration, Fig. 8 shows the potential $E(x,y)=x^4/8-x^3+2x^2+y^4/5+6\exp[-2(x-2)^2-10y^2]$ which has two minima connected by two paths. A subsampling of the data set generated by Monte Carlo simulation using this potential is shown in Fig. 9 (left panel) with the corresponding diffusion map shown in the right panel of the figure. For this data set $\mathbf{X}=[x,y]$; to compute the diffusion map we use $\alpha=0$, and $\sigma=0.5$ in Eq. (4.1). Figure 9 (right panel) shows that points close to the bottom of the wells are mapped to tight clusters in the diffusion map, with a clear distinction between data points on each of the two transition pathways between the minima.

[1] D. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[2] D. Adalsteinsson, D. McMillen, and T. Elston, BMC Bioinf. **5**, 1 (2004).
[3] D. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[4] R. Erban, I. Kevrekidis, D. Adalsteinsson, and T. Elston, J. Chem. Phys. **124**, 084106 (2006).
[5] H. Salis and Y. Kaznessis, J. Chem. Phys. **122**, 054103 (2005).
[6] C. Rao and A. Arkin, J. Chem. Phys. **118**, 4999 (2003).
[7] Y. Cao, D. Gillespie, and L. Petzold, J. Chem. Phys. **122**, 14116 (2005).
[8] E. Haseltine and J. Rawlings, J. Chem. Phys. **117**, 6959 (2002).
[9] W. E. D. Liu and E. Vanden-Eijnden, J. Chem. Phys. **123**, 194107 (2005).
[10] S. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis, J. Chem. Phys. **122**, 024112 (2005).
[11] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, Proc. Natl. Acad. Sci. U.S.A. **102**, 7426 (2005).
[12] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, Proc. Natl. Acad. Sci. U.S.A. **102**, 7432 (2005).
[13] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, Appl. Comput. Harmon. Anal. **21**, 113 (2006).
[14] I. Kevrekidis, C. Gear, J. Hyman, P. Kevrekidis, O. Runborg, and K. Theodoropoulos, Commun. Math. Sci. **1**, 715 (2003).
[15] M. Haataja, D. Srolovitz, and I. Kevrekidis, Phys. Rev. Lett. **92**, 160603 (2004).
[16] G. Hummer and I. Kevrekidis, J. Chem. Phys. **118**, 10762 (2003).
[17] S. Sriraman, I. Kevrekidis, and G. Hummer, Phys. Rev. Lett. **95**, 130303 (2005).
[18] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, *Advances in Neural Information Processing Systems*, edited by Y. Weiss, B. Schlösskopf, and J. Platt (MIT Press, Cambridge, MA, 2006), Vol. 18, pp. 955–962.
[19] M. Rathinam, L. Petzold, Y. Cao, and D. Gillespie, J. Chem. Phys. **119**, 12784 (2003).
[20] Y. Cao, D. Gillespie, and L. Petzold, J. Comput. Phys. **206**, 395 (2005).
[21] T. Gardner, C. Cantor, and J. Collins, Nature (London) **403**, 339 (2000).
[22] J. Hasty, D. McMillen, and J. Collins, Nature (London) **420**, 224 (2002).
[23] T. Kepler and T. Elston, Biophys. J. **81**, 3116 (2001).
[24] H. Risken, *The Fokker-Planck Equation, Methods of Solution and Applications* (Springer-Verlag, Berlin, 1989).
[25] D. Kopelevich, A. Panagiotopoulos, and I. Kevrekidis, J. Chem. Phys. **122**, 044908 (2005).
[26] C. Siettos, M. Graham, and I. Kevrekidis, J. Chem. Phys. **118**, 10149 (2003).
[27] M. Belkin and P. Niyogi, Neural Comput. **15**, 1373 (2003).
[28] R. Lehoucq, D. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods* (SIAM, Philadelphia, PA, 1998).
[29] C. Baker, *The Numerical Treatment of Integral Equations* (Clarendon, Oxford, 1977).
[30] Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet, Neural Comput. **16**, 2197 (2004).
[31] S. Kirkpatrick, C. Gelatt, and M. Vecchi, Science **34**, 671 (1983).
[32] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes* (Cambridge University Press, Cambridge, 1992).
[33] J. P. Ryckaert, G. Ciccotti, and H. Berendsen, J. Comput. Phys. **23**, 327 (1977).
[34] R. Erban, I. Kevrekidis, and H. Othmer, Physica D **215**, 1 (2006).
[35] G. M. Torrie and J. P. Valleau, Chem. Phys. Lett. **28**, 578 (1974).
[36] C. Gear and I. Kevrekidis, Phys. Lett. A **321**, 335 (2004).
[37] T. Frewen, I. Kevrekidis, and G. Hummer (unpublished).
[38] F. Chung, A. Grigor'yan, and S. Yau, Commun. Anal. Geom. **8**, 969 (2000).
[39] J. Tenenbaum, V. de Silva, and J. Langford, Science **290**, 2319 (2000).
[40] S. Roweis and L. Saul, Science **290**, 2323 (2000).
[41] D. Donoho and C. Grimes, Proc. Natl. Acad. Sci. U.S.A. **100**, 5591 (2003).