JCTC Journal of Chemical Theory and Computation

Enhanced Modeling via Network Theory: Adaptive Sampling of Markov State Models

Gregory R. Bowman,[†] Daniel L. Ensign,[‡] and Vijay S. Pande*[,†,‡]

Biophysics Program and Department of Chemistry, Stanford University, Stanford, California 94305

Received November 22, 2009

Abstract: Computer simulations can complement experiments by providing insight into molecular kinetics with atomic resolution. Unfortunately, even the most powerful supercomputers can only simulate small systems for short time scales, leaving modeling of most biologically relevant systems and time scales intractable. In this work, however, we show that molecular simulations driven by adaptive sampling of networks called Markov State Models (MSMs) can yield tremendous time and resource savings, allowing previously intractable calculations to be performed on a routine basis on existing hardware. We also introduce a distance metric (based on the relative entropy) for comparing MSMs. We primarily employ this metric to judge the convergence of various sampling schemes but it could also be employed to assess the effects of perturbations to a system (e.g., determining how changing the temperature or making a mutation changes a system's dynamics).

1. Introduction

Molecular dynamics simulations are a powerful means of understanding both the thermodynamics and kinetics of molecular processes like protein folding and conformational changes. Unfortunately, such processes are highly sensitive to the underlying chemical details. For example, point mutations in the amino acid sequence of a protein may have significant effects on its kinetics,[1] and a small number of point mutations can even drastically change the native structure.[2] Thus, atomistic simulations are required to make quantitative connections with experiments.[3,4]

Advances in computing have made it possible to rapidly generate huge data sets even at this level of chemical detail;[5,6] however, these data sets are still insufficient. A typical computer can only simulate ∼5 ns/day of protein folding and would thus take over 500 years to simulate one millisecond, an average folding time typical of proteins. Whether one is interested in dynamics or merely equilibrium probabilities, a kinetic perspective on this problem that explicitly considers the rate of equilibration reveals that metastability, or the presence of long-lived states that act as "traps", is a common source of inefficiency.

One approach to dealing with this issue is to make tremendous investments in specialized software and hardware for generating long simulations.[7] While theoretically sound,[8] this serial approach often only results in simulations that are long *relative* to standard trajectories. However, a *truly long* simulation must be orders of magnitude longer than the slowest relaxation time so that the probabilities of all states and pathways can be estimated accurately. Even if such a simulation were possible, the task of analyzing the data would still remain.[7,9] Moreover, serial approaches are inherently inefficient, both due to parallelization overhead and, more importantly, the fact that they waste hundreds of years of computing time waiting for rare events.

A statistical approach provides a fundamentally different perspective on model construction. Rather than attempting to generate one realization of an entire process, one instead aims to generate an ensemble of events in parallel. For example, a number of methods have been developed for exploiting statistical mechanics to simulate protein folding more efficiently.[10−13] Most of these approaches rely on the fact that, in two-state protein folding, the waiting time for observing a transition is exponentially distributed but the actual transition times are quite rapid.[14] Thus, proteins often

* Corresponding author e-mail: pande@stanford.edu.

[†] Biophysics Program.

[‡] Department of Chemistry.

10.1021/ct900620b © 2010 American Chemical Society
Published on Web 02/17/2010

fold much faster or slower than the average folding time. Such approaches are amenable to commodity hardware and take far less wall-clock time than a serial approach with an equivalent amount of sampling, particularly when combined with grid computing.[5] Unfortunately, these methods are generally only applicable to two-state systems and may require simulations of an unknown minimum length.[15] Some multistate generalizations exist[16] but quickly become computationally intractable.

Markov state models (MSMs) extend this work by allowing for a tractable, multistate scheme that allows efficient modeling of any system exhibiting metastability.[17] A MSM is a network with nodes corresponding to metastable states and edges describing the rates of transitioning between pairs of states, akin to a map with cities connected by roads labeled with speed limits. Rather than attempting to generate one realization of an entire process, one can exploit the decomposition of conformational space into multiple metastable states to gather statistics on each step of the process independently, allowing a problem to be broken up into more manageable and trivially parallelizable pieces.

Mathematically, MSMs are represented as transition probability matrices, with the entry in row i and column j giving the probability of transitioning from state i to state j within a time interval called the lag time of the model. Building MSMs is a challenging task, but significant progress has been made over the past few years,[18−21] leading to freely available software for automatically constructing these models.[18] While MSMs could be used to analyze truly long simulations, their ultimate value lies in their ability to facilitate efficient model construction by allowing precise, parallel determination of the transition rates between states by running many short simulations from each of them.

Adaptive sampling algorithms for MSM construction take this statistical approach a step further.[22−24] In adaptive sampling, one first obtains an initial model of the entire process of interest by any means possible. One then iteratively calculates the contribution of each step of the process to uncertainties in some observable of interest via Bayesian statistics and runs numerous parallel simulations of the steps that can lead to the greatest increases in precision until the desired level of statistical certainty is achieved. Such an approach was recently shown to lead to dramatic reductions in the statistical uncertainty in the observable of interest relative to other refinement schemes.[22]

However, a number of important questions remain to be answered. First, does adaptive sampling improve the global model quality or just local components that are important for the observable of interest? Exactly how much more efficient is adaptive sampling? And finally, is adaptive sampling capable of discovering previously unknown components of a model, or is it only able to refine the initial model it is given?

In this work, we address these questions using a MSM for the villin headpiece (HP-35 NleNle) that was recently constructed from atomistic simulations with explicit solvent.[19] We then move on to simple models, where the role of the network is clear, to gain an intuition for our results and test whether such methods could be more broadly applicable to a wide class of different types of systems. These analyses rely on a new distance metric for MSMs developed in section 2.2, which should prove generally useful for evaluating various sampling schemes and even assessing the effects of perturbations to a system (like changes in temperature or even mutations).

2. Theoretical Underpinnings

2.1. Adaptive Sampling. In adaptive sampling approaches to MSM construction, simulations are run iteratively to minimize uncertainties in some property of a model.[22−24] In this work, adaptive sampling is performed as follows:

(1) Perform N simulations of L steps starting from a particular starting state(s).

(2) Build a MSM only including those states identified so far.

(3) Calculate the contribution of each state to uncertainty in the slowest kinetic rate following ref 22.

(4) Start N new simulations of L steps distributed among the states in proportion to their contribution to uncertainty in the slowest rate.

(5) Repeat steps 2−4 for some number of iterations.

All the MSMs in this work were constructed and analyzed with the MSMBuilder package (which is freely available at https://simtk.org/home/msmbuilder/)[18] modified such that transition count matrices were not symmetrized by counting the transitions that would have been observed if one watched each simulation backward.

We note that, in the past, simulations in each round of adaptive sampling were all started from the same initial state (the one contributing most to uncertainty in the quantity of interest).[22] The intuition behind our alteration was that, as the number of simulations (N) becomes large, starting all the simulations from one state would be excessive as fewer would be sufficient to drastically reduce the uncertainty. Instead, it would be preferable to allocate some of these excess simulations to reduce uncertainties in other states' transition probabilities. Indeed, we have found that our modified procedure yields better results for sufficiently large N on reasonably complex networks and gives equivalent results for simple networks and small N.

To demonstrate the utility of this algorithm, we carried out adaptive sampling with synthetic trajectories generated from transition count matrices. To generate synthetic simulations from a transition count matrix, we first normalize each row to obtain a transition probability matrix. At each time step (or each lag time), the next state is chosen according to the distribution of transition probabilities for the current state. The prior described below is not used for these calculations, so the matrices used to generate trajectories tend to be sparse.

2.2. Quantifying the Similarity between MSMs. In order to monitor the convergence of any sampling scheme, it is important to first develop a similarity metric that is capable of measuring the global quality of a test model relative to some reference model. Such a metric would also have broad usefulness, as there are several reasons for comparing MSMs quantitatively. For example, this metric could be used to compare MSMs generated by two different simulation methods, allowing one to directly compare the resulting dynamics. Alternatively, one could compare MSMs generated by two somewhat different, but related systems, such as comparing the simulations of the dynamics of two point mutants of a given protein.

We have developed such a distance metric for MSMs that is based on the relative entropy, which is a common measure of the distance between two probability distributions in information theory[25] with important physical implications.[26] The relative entropy between two normalized distributions P and Q, over a common set of outcomes, is

$$D(P||Q) = \sum_i P_i \log \frac{P_i}{Q_i}$$

where P_i is the probability of outcome i, P is a reference distribution, and Q is some test distribution.

A MSM consists of one normalized distribution per state, which gives the probability of transitioning to each other state within one lag time. We define the relative entropy between a reference and test MSM, with transition matrices P and Q respectively, as

$$D(P||Q) = \sum_{i,j}^{N} P_i P_{ij} \log \frac{P_{ij}}{Q_{ij}} \tag{1}$$

where P_i is the equilibrium probability of state i, P_{ij} is the probability of transitioning from state i to state j during one lag time, and N is the number of states. Intuitively, our relative entropy metric is the sum of the relative entropies between the transition probability distributions for each state weighted by their stationary probabilities.

One may derive our relative entropy metric for MSMs more formally by considering that the entropy (H) of a sample path of a stochastic process, normalized by its length, is also called the entropy rate. An important theorem in information theory is the following:

Theorem. For an ergodic stochastic process, $X_1, ..., X_n$

$$\lim_{n\to\infty} \frac{1}{n} H(X_1, ..., X_n) = \lim_{n\to\infty} H(X_n|X_1, ..., X_{n-1})$$

For a Markov Chain, the right-hand side takes a very simple form, because the conditional entropy only depends on the previous step, which converges to the stationary distribution.

In the following, we prove a similar statement for the relative entropy between the paths of two Markov chains as n goes to infinity. For two Markov chains p and q with state space Ω, we would like to compute:

$$\lim_{n\to\infty} \frac{1}{n} D(p(X_1, ..., X_n)||q(X_1, ..., X_n))$$

For simplicity, let us define lowercase $x_n = \{X_1, ..., X_\mathrm{n}\}$. Then, by the chain rule for the relative entropy, we get

$$\lim_{n\to\infty} \frac{1}{n}[D(p(x_{n-1})||q(x_{n-1})) + D(p(X_n|x_{n-1})||q(X_n|x_{n-1}))] \tag{2}$$

Equation 2.65 in Cover and Thomas[27] defines the conditional relative entropy above as the expectation of the relative entropy between the conditional distributions of X_n given x_{n-1}, with respect to the distribution of x_{n-1}. This means that

$$\begin{aligned} D(p(X_n|x_{n-1})||q(X_n|x_{n-1})) &= \sum_{y\in\Omega^{n-1}} p(x_{n-1} = y) D(p(X_n|y)||q(X_n|y)) \\ &= \sum_{Y\in\Omega} p(X_{n-1} = Y) D(p(X_n|Y)||q(X_n|Y)) \end{aligned}$$

where we have grouped terms with the same final state in the “history” y, which have the same relative entropy factor, and summed their probabilities to obtain the marginal probability over $X_{n-1.}$

Repeating the step that led to eq 2 many times yields

$$\lim_{n\to\infty} \frac{1}{n}[\sum_{m=2}^{n} D(p(X_m|x_{m-1})||q(X_m|x_{m-1}))] + D(p(X_1)||q(X_1))$$

If the initial state is deterministic, the last term is just zero. As for the first term, as n goes to infinity, the distribution of X_{m-1} goes to the stationary distribution of p, which we call μ. Then, using the equation for the conditional entropy,

$$\lim_{n\to\infty} D\left(p(X_n\middle|x_{n-1})\middle|\,|q(X_n\middle|x_{n-1})\right) = \sum_{Z\in\Omega} \mu(Z) \sum_{Y\in\Omega} p(Y|Z) \log\left[\frac{p(Y|Z)}{q(Y|Z)}\right]$$

Since the terms in the series converge to a limit, their Cesaro means converge to the same limit, so

$$\lim_{n\to\infty} \frac{1}{n} D(p(X_1, ..., X_n)||q(X_1, ..., X_n)) = \sum_{Z\in\Omega} \mu(Z) \sum_{Y\in\Omega} p(Y|Z) \log\left[\frac{p(Y|Z)}{q(Y|Z)}\right]$$

The terms $p(Y|Z)$ and $q(Y|Z)$ are just the elements of the transition matrices of p and q, respectively; so this is equivalent to eq 1.

2.3. Prior for Relative Entropy and Adaptive Sampling. There is always some probability of transitioning between every pair of states, though these probabilities may be low enough that no actual transitions are observed. To account for this, as well as to reflect our lack of prior knowledge about the transition probabilities, we add a pseudocount of $1/N$ to every element of the transition count matrix, where N is the number of states, before normalizing each row to find the transition probability matrix, as in refs 22 and 28. The intuition behind this choice is that for a state to exist we must observe at least one count in that state, but before observing any real data the probability of this count leading to any other state is equal. From a Bayesian perspective, these pseudocounts equate to a uniform prior. These pseudocounts also prevent the relative entropy metric from becoming infinite whenever a zero is encountered in a MSM's transition probability matrix. It is often the case that certain transitions are not observed, so this correction is of great practical importance.

2.4. Villin Simulations and MSM. The simulation details for the original ~450 villin simulations are described in detail in ref 29. In short, ~450 constant temperature molecular dynamics simulations with explicit solvent and up to 2 μs in length were run from nine initial configurations drawn from high temperature unfolding simulations at 373 K. Ref 19 describes the construction of a 10 000 microstate MSM that faithfully reproduces the raw simulation data. For the purposes of this work, we lumped these 10 000 microstates into 500 macrostates exhibiting metastability and having an equivalent Markov time (15 ns). This lumping was done with the MSMBuilder package.[18] The macrostates containing the nine initial configurations used during the real simulations were used as the starting points for adaptive sampling. Simulations of just 30 ns were used for adaptive sampling.

2.5. Simple Models. The transition count matrices for simple models S and P (C_S and C_P respectively) are

$$C_S = \begin{bmatrix} 6000 & 3 & 0 & 0 & 0 & 0 \\ 3 & 1000 & 3 & 0 & 0 & 0 \\ 0 & 3 & 1000 & 3 & 0 & 0 \\ 0 & 0 & 3 & 1000 & 3 & 0 \\ 0 & 0 & 0 & 3 & 1000 & 3 \\ 0 & 0 & 0 & 0 & 3 & 90\,000 \end{bmatrix}$$

and

$$C_P = \begin{bmatrix} 6000 & 2 & 2 & 0 & 0 & 0 \\ 2 & 1000 & 0 & 2 & 2 & 0 \\ 2 & 0 & 1000 & 2 & 2 & 0 \\ 0 & 2 & 2 & 1000 & 0 & 2 \\ 0 & 2 & 2 & 0 & 1000 & 2 \\ 0 & 0 & 0 & 2 & 2 & 90\,000 \end{bmatrix}$$

where the entry in row i and column j gives the number of transitions observed from state i to state j.

Mean first passage times were calculated following ref 28. The mean first passage times for S and P are ~13 000 and ~5000 steps, respectively. Other equilibrium properties can be obtained by normalizing each row to obtain a transition probability matrix and then solving for the eigenvalues and eigenvectors of this matrix. For example, normalizing the first eigenvector (e.g., the one corresponding to an eigenvalue of 1) gives the equilibrium probabilities of each state. Subsequent eigenvalue/eigenvector pairs give kinetic rates and the states involved in these transitions, respectively.[17] Once again, the MSMBuilder package[18] was used for analysis of these models.

Plots of the average relative entropy as a function of simulation number and length were generated by running 600 simulations of 5000 steps for each model. Average relative entropies over 10 random samples of N trajectories from this pool were then calculated and plotted. Similar plots for our adaptive sampling scheme were also generated by averaging over 10 independent runs.

3. Results and Discussion

3.1. Application to Villin MSM. With these tools in place, we are now in a position to assess the efficacy of adaptive sampling using a previously calculated MSM for the villin headpiece[19] as a model system. In particular, we would like to assess two types of efficiency. First, given our desire to push the envelope of what is possible in a reasonable amount of time, can adaptive sampling reduce the wall-clock time necessary to achieve a given model quality? Second, given our desire to mitigate negative impacts on the environment, can adaptive sampling reduce the amount of resources (in this case computer time) necessary to achieve a given model quality?

To address these questions, we have performed adaptive sampling with a variable number of simulations per iteration generated from our villin MSM. We then assume each simulation progresses at a rate of 5 ns/day, a typical value for modern personal computers, and compare the convergence of our adaptive simulations to the gold-standard model from ref 19 (that was validated by comparison to both the raw simulation data and experiments) with the convergence of a single long reference simulation to the same gold standard. Convergence to the gold-standard model is measured with our relative entropy metric for MSMs (described in section 2.2).

Figure 1A shows that the wall-clock time efficiency of adaptive sampling scales linearly up to 5000 simulations per iteration. That is, adaptive sampling with N simulations per iteration can reduce the wall-clock time necessary to achieve a given model quality by a factor of N for N as high as 5000. Using more simulations will help but will only reduce the wall-clock time by a factor of αN, where $\alpha < 1$. The crucial result, however, is that one can reduce a calculation that would take decades to run with traditional methods to a calculation that can be run in a matter of days with adaptive sampling.

Adaptive sampling can also greatly reduce the resource requirements for achieving a given model quality. For example, Figure 1B shows the computer time necessary to achieve a given model quality for one long simulation and adaptive sampling with a varying number of simulations per iteration. This figure shows that adaptive sampling requires about half as much computer time to achieve the same model quality as one long simulation. Once again, the relative efficiency of adaptive sampling begins to fall off beyond some optimal number of simulations per iteration.

3.2. Application to Simple Models. To gain intuition for the applicability of adaptive sampling to other systems, we have also applied it to two classic network topologies, shown in Figure 2A and defined more thoroughly in section 2.5. These models are representative of problems with metastability; their equilibrium properties can be derived analytically and used as an unambiguous reference, and truly long simulations are feasible.

Both models have states with approximately the same equilibrium and transition probabilities, such that differences between their behaviors can be attributed to differences between their topologies. More specifically, states 1−6 have equilibrium populations of 6%, 1%, 1%, 1%, 1%, and 90%, respectively. Drawing an analogy to protein folding, state 1 is the unfolded state, state 6 is the folded state, and the remaining states are intermediates. Thus, S has a single folding pathway, and P has parallel folding pathways.

The reduced connectivity in S results in longer time scale transitions relative to P. In fact, the mean first passage time

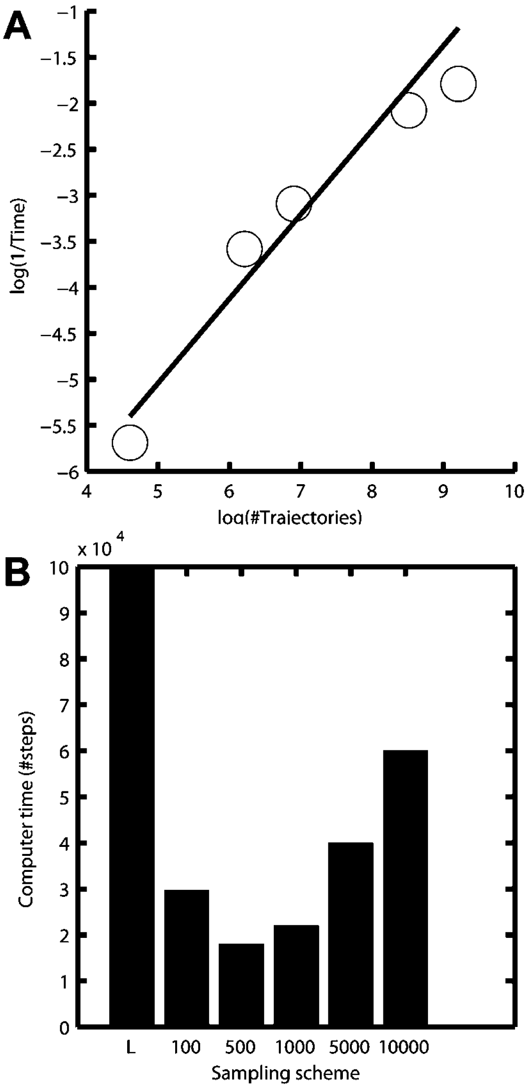

Figure 1. Scaling for adaptive sampling of villin as the number of parallel simulations (*N*) used during each round is varied. (A) Wall-clock time scaling as *N* is varied. The black line is a best fit to the linear portion of the data (circles), which extends up to 5000 simulations per iteration. (B) Computer time required to achieve a given model quality (relative entropy) for various sampling schemes. *L* refers to one long trajectory, and the numbers refer to the number of parallel simulations used in each iteration of adaptive sampling. All results come from averaging over 10 independent runs. Each step equates to 15 ns.

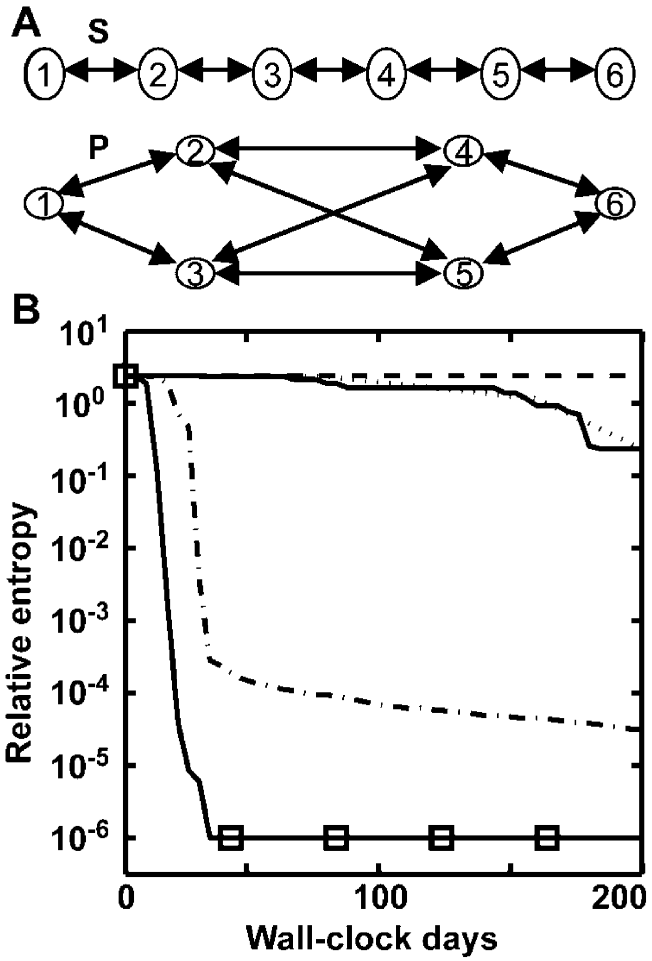

Figure 2. (A) The two models, S and P. (B) Distance from the true model (measured via the relative entropy) as a function of wall-clock time for adaptive sampling versus one long simulation of S (assuming 5 steps/day to mimic 5 ns/day in protein folding simulations). The lines are one long simulation (dashed line) and adaptive sampling with 10 simulations of 20 steps (solid line), 10 simulations of 200 steps (dotted line), 100 simulations of 20 steps (dash-dot line), and 1000 simulations of 20 steps (black squares) per iteration.

(MFPT) between states 1 and 6 is about three times longer in S than in P, making S considerably harder to sample. In addition, such linear models are often cited as a case where the holistic, long-trajectory approach is absolutely necessary; nevertheless, adaptive sampling is able to learn the network more efficiently than traditional approaches, as shown in Figure 2B. This figure shows how close various schemes can approach the true model for S given a set amount of wall-clock time and starting from state 1 to mimic the practice of starting protein folding simulations from an arbitrary conformation in the unfolded state.

To provide some intuition for our distance metric, Figure 3 shows the evolution of the relative entropy and the estimated free energy of each state in S during adaptive sampling. Adaptive sampling was carried out by running 10 simulations from state 1 and then repeatedly building a MSM and starting 10 new simulations from the state contributing most to uncertainty in the slowest process. Small jumps in the relative entropy are found each time a state with a low population is discovered (or, equivalently, when a new path is discovered for this model), and a very large jump is evident when the most populated state, state 6, is discovered. Slow decay occurs between these jumps. Thus, our metric is most sensitive to state and path discovery but still captures improvements in estimates of the transition probabilities along known paths. Such behavior is desirable as models that miss important states or paths should be penalized more than ones with imperfect transition probabilities.

Figure 4 shows a more thorough comparison of adaptive sampling and reference simulations with an equal amount of sampling for various numbers and lengths of simulations. Evaluation of the reference simulations for both S and P demonstrates that achieving a reasonable model quality by naively starting simulations from state 1 requires simulations

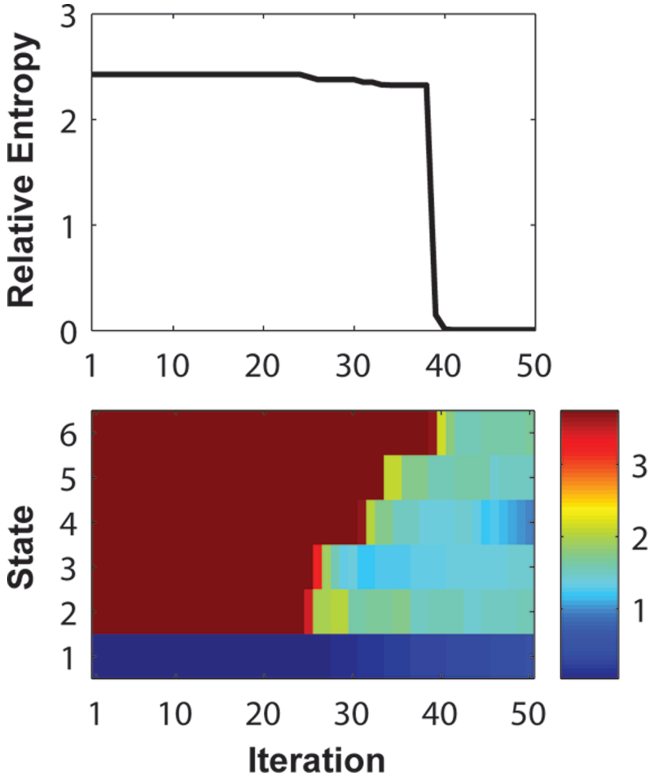

Figure 3. Relative entropy (top) and free energy of each state in kcal/mol (bottom) as a function of the adaptive sampling iteration on model S.

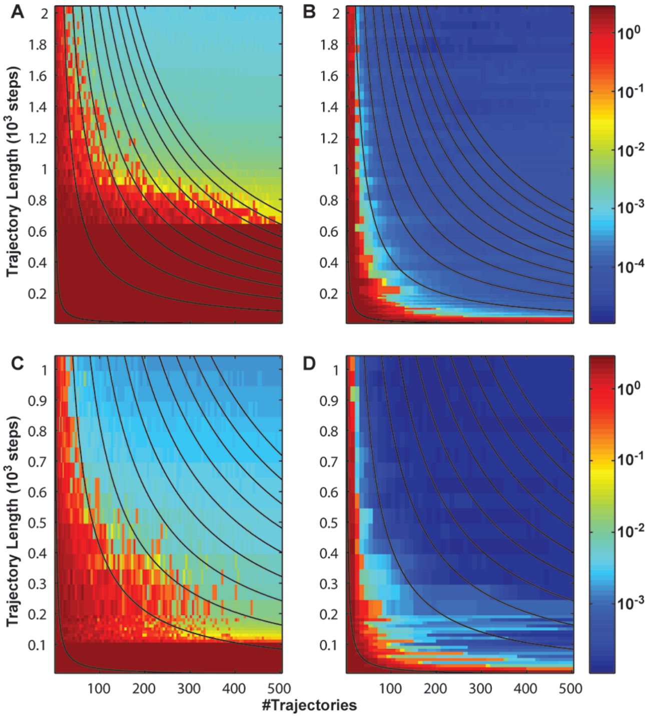

Figure 4. Distance from the true model (measured via the relative entropy) as a function of the number and length of simulations averaged over 10 independent samples. (A) Reference distribution for S, (B) adaptive sampling of S, (C) reference distribution for P, and (D) adaptive sampling of P. All simulations for the reference distributions started from state 1. The first 10 simulations for adaptive sampling started from state 1, and subsequent batches of simulations started from the state contributing most to uncertainty in the slowest process. Black lines are contours of equal amounts of data.

of some minimal length, though this minimal length is shorter for P than S in terms of the absolute number of steps. Moreover, adaptive sampling is able to gain valuable information from much shorter and fewer simulations regardless of the topology of the network, that is, whether there is a single folding pathway or multiple pathways. This figure also shows that adaptive sampling generally benefits from using more parallel simulations but not longer ones. An important point is that each data point in Figure 4B and D depends on the data points to its left. For example, to fill in the row corresponding to simulations of length 100, 10 independent adaptive sampling runs of 50 iterations were performed. The first round of each adaptive sampling run was used to compute average relative entropies for 1−10 simulations, the first and second round of each run (which depends on the first round) for 11−20 simulations, and so forth. As a result, there is some horizontal streakiness in these figures. We also note that adaptive sampling results in smaller uncertainties in the relative entropies shown in Figure 4 (see Figures S1 and S2, Supporting Information).

Finally, we find that the scaling of adaptive sampling of our simple networks is similar to that found for villin, as shown in Figure 5. One noteworthy difference is that our simple models saturate (i.e., fall short of linear scaling as additional parallel simulations are run) earlier than villin. Comparison of the two simple models also shows that S saturates before P. For S, adaptive sampling scales linearly up to 150 parallel simulations. For P, adaptive sampling scales linearly up to 500 simulations. The improved scaling for P is the result of the increased complexity of the network topology of P compared to S. Each node in P has more connections to learn, and the algorithm benefits from doing this in parallel. Indeed, the complexity of our villin model is much greater than either of these simple networks, and as discussed previously, villin scales linearly up to 5000 simulations per iteration. Thus, we expect that we can achieve linear scaling well beyond 5000 simulations per iteration for systems that are more complex than the villin MSM that we sampled from.

3.3. Applicability. The adaptive sampling algorithm employed here was developed for application to MSMs with metastable states. That is, it assumes that every state has a self-transition probability greater than 0.5 such that a simulation in one state is more likely to stay there than to transition to a new state. This property helps to ensure a separation of time scales (fast intrastate transitions, slow interstate transitions) and, therefore, that the model is Markovian because a simulation can lose memory of its previous state before transitioning to a new one. Thus, the procedure for *ab initio* adaptive sampling is (1) run some initial simulations, (2) cluster all the simulation data into microstates, (3) lump these microstates into metastable macrostates, (4) calculate the contribution of each macrostate to uncertainties in the slowest rate (or some other observable), (5) start new simulations from each state in proportion to its contribution to the overall uncertainty, and (6) repeat steps 2−5 until the desired level of statistical certainty is achieved.

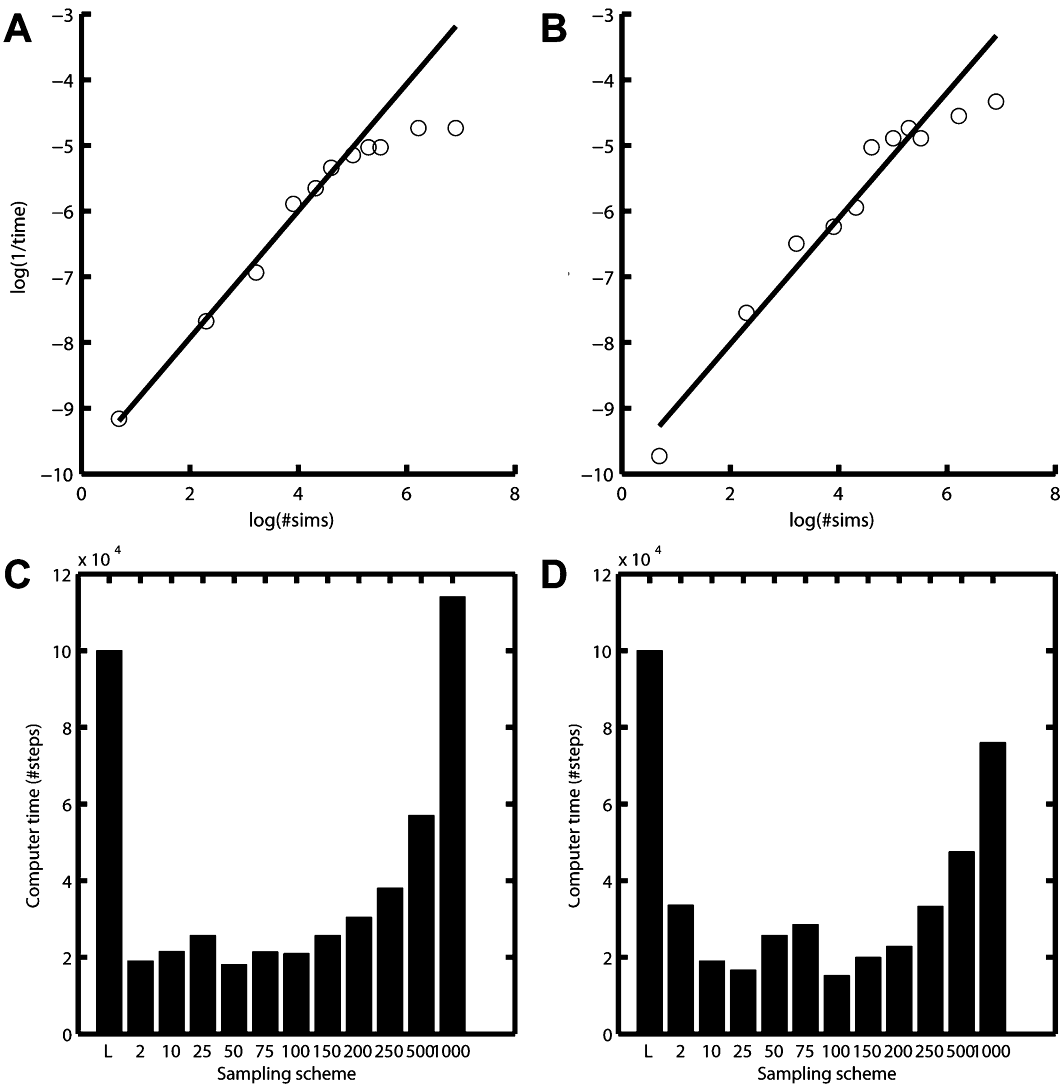

Figure 5. Scaling for adaptive sampling of our simple models as the number of parallel simulations (*N*) used during each round is varied. (A and B) Wall-clock time scaling as *N* is varied for simple models S and P, respectively. The black line is a best fit to the linear portion of the data (circles). (C and D) Computer time required to achieve a given model quality (relative entropy) for various sampling schemes applied to S and P, respectively. *L* refers to one long trajectory, and the numbers refer to the number of parallel simulations used in each iteration of adaptive sampling. All results come from averaging over 10 independent runs.

In the future, it will be interesting to explore whether this adaptive sampling algorithm is equally applicable to more fine grained divisions of conformational space (e.g., at the microstate level) as the lumping stage would no longer be necessary. In addition, recent work has shown that more fine grained MSMs are better for obtaining quantitative predictions of experimental observables,[19,30,31] so it could be advantageous to do refinement at this level.

The relative entropy metric assumes that the two models being compared have the same state space. Comparing two simulation data sets therefore requires the following steps: (1) define a state space common to both data sets (i.e., by using both data sets for clustering to define microstates and, optionally, lumping to define macrostates), (2) compute transition probability matrices for each data set independently, and (3) compute the relative entropy between these matrices.

4. Conclusions

Together, our results with villin and fundamental model systems demonstrate the tremendous value of adaptive sampling. Since model quality has been assessed with a global metric and shows strong agreement between adaptive sampling results and the true model, we can conclude that adaptive sampling to minimize uncertainties in the slowest kinetic rate improves the global quality of a model. Moreover, adaptive sampling is significantly more efficient than a single long simulation, both in terms of the wall-clock time and resources required to achieve a given model quality, up to some saturation point. In fact, adaptive sampling with *N* parallel simulations requires about a factor of 2 less computer-time and a factor of *N* less wall-clock time. Considering that *N* can easily be as large as 10 000 (or more),[5] this can be a truly dramatic advantage in wall-clock time, turning calculations normally requiring decades into routine calculations on the time scale of days. Finally, since our simulations started from just a couple of states, we can conclude that adaptive sampling is capable of discovering new model components *given no prior knowledge of the system* and is thus useful for model construction in addition to model refinement.

The adaptive sampling method described here may be directly applied to learn models from simulations of metastable phenomena, leading to significant resource and time savings in fields like molecular and quantum mechanics, but is not limited to these applications. Given a means to prepare samples within a given state, it could be applied equally well to experimental techniques, such as single molecule FRET and force extension experiments. More broadly, minimizing uncertainties in a model is likely to prove valuable even when metastability is not present. Similar methods may also be useful for understanding other complex network dynamics, as in signaling pathways.

Acknowledgment. Thanks to Sergio Bacallado for help with the relative entropy metric. This work was funded by NIH R01-GM062868 and NIH U54 GM072970. G.R.B. was supported by the NSF GRFP.

Supporting Information Available: Figures S1 and S2. This material is available free of charge via the Internet at http://pubs.acs.org.

References

(1) Liu, F.; Du, D.; Fuller, A. A.; Davoren, J. E.; Wipf, P.; Kelly, J. W.; Gruebele, M. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105*, 2369–2374.

(2) He, Y.; Yeh, D. C.; Alexander, P.; Bryan, P. N.; Orban, J. *Biochemistry* **2005**, *44*, 14055–14061.

(3) Rhee, Y. M.; Pande, V. S. *J. Chem. Phys.* **2006**, *323*, 66–77.

(4) Bradley, P.; Misura, K. M.; Baker, D. *Science* **2005**, *309*, 1868–1871.

(5) Shirts, M.; Pande, V. S. *Science* **2000**, *290*, 1903–1904.

(6) Das, R.; Qian, B.; Raman, S.; Vernon, R.; Thompson, J.; Bradley, P.; Khare, S.; Tyka, M. D.; Bhat, D.; Chivian, D.; Kim, D. E.; Sheffler, W. H.; Malmstrom, L.; Wollacott, A. M.; Wang, C.; Andre, I.; Baker, D. *Proteins* **2007**, *69*, 118–128.

(7) Klepeis, J. L.; Lindorff-Larsen, K.; Dror, R. O.; Shaw, D. E. *Curr. Opin. Struct. Biol.* **2009**, *19*, 120–127.

(8) Geyer, C. J. *Stat. Sci.* **1992**, *7*, 473–511.

(9) King, R. D.; Rowland, J.; Oliver, S. G.; Young, M.; Aubrey, W.; Byrne, E.; Liakata, M.; Markham, M.; Pir, P.; Soldatova, L. N.; Sparkes, A.; Whelan, K. E.; Clare, A. *Science* **2009**, *324*, 85–89.

(10) Pande, V. S.; Baker, I.; Chapman, J.; Elmer, S. P.; Khaliq, S.; Larson, S. M.; Rhee, Y. M.; Shirts, M. R.; Snow, C. D.; Sorin, E. J.; Zagrovic, B. *Biopolymers* **2003**, *68*, 91–109.

(11) Bolhuis, P. G.; Chandler, D.; Dellago, C.; Geissler, P. L. *Annu. Rev. Phys. Chem.* **2002**, *53*, 291–318.

(12) Faradjian, A. K.; Elber, R. *J. Chem. Phys.* **2004**, *120*, 10880–10889.

(13) Shirts, M. R.; Pande, V. S. *Phys. Rev. Lett.* **2001**, *86*, 4983–4987.

(14) Chung, H. S.; Louis, J. M.; Eaton, W. A. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106*, 11837–11844.

(15) Fersht, A. R. *Proc. Natl. Acad. Sci. U. S. A.* **2002**, *99*, 14122–14125.

(16) Rogal, J.; Bolhuis, P. G. *J. Chem. Phys.* **2008**, *129*, 224107.

(17) Schutte, C. Ph.D. Thesis, Freie Universitat, Berlin, 1999.

(18) Bowman, G. R.; Huang, X.; Pande, V. S. *Methods* **2009**, *49*, 197–201.

(19) Bowman, G. R.; Beauchamp, K. A.; Boxer, G.; Pande, V. S. *J. Chem. Phys.* **2009**, *131*, 124101.

(20) Chodera, J. D.; Singhal, N.; Pande, V. S.; Dill, K. A.; Swope, W. C. *J. Chem. Phys.* **2007**, *126*, 155101.

(21) Noe, F.; Fischer, S. *Curr. Opin. Struct. Biol.* **2008**, *18*, 154–162.

(22) Hinrichs, N. S.; Pande, V. S. *J. Chem. Phys.* **2007**, *126*, 244101.

(23) Roblitz, S. Ph.D. Thesis, Freie Universitat, Berlin, 2008.

(24) Huang, X.; Bowman, G. R.; Bacallado, S.; Pande, V. S. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106*, 19765–19769.

(25) MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms*; Cambridge University Press: Cambridge, U. K., 2003.

(26) Shell, M. S. *J. Chem. Phys.* **2008**, *129*, 144108.

(27) Cover, T. M.; Thomas, J. A. *Elements of Information Theory*, 2nd ed.; Wiley-Inter Science: Hoboken, NJ, 2006.

(28) Singhal, N.; Pande, V. S. *J. Chem. Phys.* **2005**, *123*, 204909.

(29) Ensign, D. L.; Kasson, P. M.; Pande, V. S. *J. Mol. Biol.* **2007**, *374*, 806–816.

(30) Sarich, M.; Noe, F.; Schutte, C. *SIAM Multiscale Model. Simul.* In submission, **2010**.

(31) Noe, F.; Schutte, C.; Vanden-Eijnden, E.; Reich, L.; Weikl, T. R. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106*, 19011–19016.

CT900620B