Stochastic Roadmap Simulation: An Efficient Representation and Algorithm for Analyzing Molecular Motion

Mehmet Serkan Apaydin* Douglas L. Brutlag† Carlos Guestrin* David Hsu‡ Jean-Claude Latombe*

Department of *Computer Science and †Biochemistry
Stanford University
Stanford, CA 94305, USA
{apaydin,guestrin,latombe}@cs.stanford.edu
brutlag@stanford.edu

‡Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599, USA
dyhsu@cs.unc.edu

Abstract

Classic techniques for simulating molecular motion, such as the Monte Carlo and molecular dynamics methods, generate individual motion pathways one at a time and spend most of their time trying to escape from the local minima of the energy landscape of a molecule. Their high computational cost prevents them from being used to analyze many pathways. We introduce *Stochastic Roadmap Simulation* (SRS), a new approach for exploring the kinetics of molecular motion by simultaneously examining multiple pathways encoded compactly in a graph, called a roadmap. A roadmap is computed by sampling a molecule's conformation space at random. The computation does not suffer from the local-minima problem encountered with existing methods. Each path in the roadmap represents a potential motion pathway and is associated with a probability indicating the likelihood that the molecule follows this pathway. By viewing the roadmap as a Markov chain, we can efficiently compute kinetic properties of molecular motion over the entire molecular energy landscape. We also prove that, in the limit, SRS converges to the same distribution as Monte Carlo simulation. To test the effectiveness of our approach, we apply it to the computation of the transmission coefficients for protein folding, an important order parameter that measures the "kinetic distance" of a protein's conformation to its native state. Our computational studies show that SRS obtains more accurate results and achieves several orders-of-magnitude reduction in computation time, compared with Monte Carlo simulation.

1. Introduction

Many interesting properties of molecular motion are best characterized statistically by considering an ensemble of pathways rather than an individual one. For example, the "new view" of protein folding kinetics replaces a single folding pathway with an energy landscape and a folding funnel [BOSW95, DC97, DK99, PGTR98]. Proteins are thought to fold in a multi-dimensional funnel by following a myriad of pathways, all leading to the native structure. To carry out computational studies of molecular motion in this framework, we need efficient algorithms that can quickly explore a large number of pathways. Unfortunately classic simulation techniques,

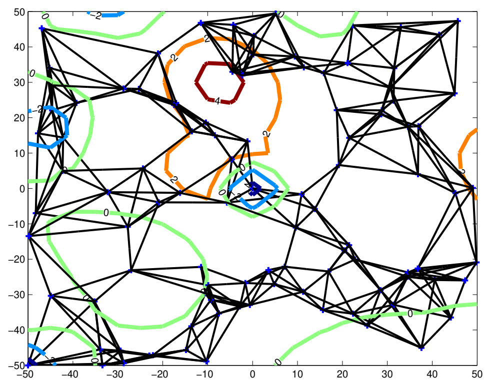

Figure 1: A probabilistic conformational roadmap superimposed on the contour plot of a hypothetical energy landscape.

such as the Monte Carlo [KW86] and molecular dynamics [Hai92] methods, generate individual pathways one at a time and waste a lot of time trying to escape from the local minima of an energy landscape. They are computationally inefficient if applied in a brute-force fashion to deal with many pathways. In this paper, we introduce *Stochastic Roadmap Simulation* (SRS), a randomized technique for sampling molecular motion and exploring the kinetics of such motion by examining multiple pathways simultaneously.

In SRS, we compactly encode many pathways in a directed graph called a *probabilistic conformational roadmap* (Figure 1), or just *roadmap* for short. Each node of the roadmap is a randomly sampled conformation of a molecule. Each (directed) edge between two nodes v_i and v_j carries a weight P_{ij}, which is the probability for the molecule to transition from v_i to v_j. Every path in the roadmap corresponds to a potential motion pathway for the molecule. A roadmap contains many pathways, with associated probabilities indicating the likelihood that the molecule may follow these pathways. In SRS, we construct a roadmap and analyze all the paths in it simultaneously to obtain kinetic information on the motion of molecules over the entire energy landscape.

To analyze a roadmap, we view molecular motion in the roadmap as a random walk on a graph. We avoid explicitly simulating the motion and obtain much of the same information by applying algebraic methods from the Markov chain theory [TK94]. Intuitively this is equivalent to performing many simulation runs simultaneously for a long time. As an example, let us consider the problem of computing the transmission coefficients for a protein in a system dominated by two stable states, a folded one and an unfolded one. The transmission coefficient τ for a conformation q is defined

TO APPEAR IN
International Conference on Research in Computational Molecular Biology (RECOMB), 2002

as the probability of reaching the folded state before the unfolded state, starting from q [DPG$^+$98]. This coefficient provides a measure of the "kinetic distance" between q and the folded state. It is possible to compute τ in a straightforward way: for every q of interest, start many of Monte Carlo simulation runs from q and count the number of times that they enter the folded state before the unfolded one [DPG$^+$98]. However, the simulation is computationally expensive, as a large number of simulation runs are required to obtain a reasonably accurate estimate of τ. With SRS, we can achieve the same result much more efficiently. This is not a surprise, because every path in a roadmap can be interpreted as a Monte Carlo simulation run. However, Monte Carlo simulation follows only one pathway at a time. It also easily gets stuck in the local minima of the energy landscape, repeatedly sampling many similar conformations without obtaining much new information. Our new approach avoids the problem by sampling directly from the space of all pathways and treating them together using algebraic methods.

SRS is a more coarse-grained simulation technique than the Monte Carlo method. The Monte Carlo method tends to focus on one pathway at a time and has a higher density of samples along that particular pathway. In contrast, SRS spreads the samples over the whole conformation space. Thus it is able to examine many motion pathways at once and extract interesting kinetic properties that are not easily accessible by other methods such as Monte Carlo simulation. In addition, we show that, in the limit, SRS and Monte Carlo simulation converge to the same sampling distribution (Section 4).

SRS is inspired by probabilistic roadmap methods for motion planning [KŠLO96]. The main idea of probabilistic roadmap methods is to construct a graph that captures the connectivity of a high-dimensional space via random sampling. Singh, *et al.* first introduced probabilistic roadmap methods to the study of molecular motion in their work on ligand-protein binding [SLB99]. These methods have since been applied to protein folding as well [ASBL01, SA01]. The earlier work treats a roadmap as a deterministic graph with heuristic edge weights based on the energy difference between molecule conformations. In contrast, we use the probabilistic conformational roadmap as a way to capture the stochastic nature of molecular motion. It enables us to exploit the knowledge from Markov chain theory to process new queries that are biologically relevant and to establish a formal relationship between SRS and Monte Carlo simulation.

The main contributions of this work are the following. SRS provides a new representation of the stochastic motion of molecules. We describe how to construct a roadmap (Section 2) and query a roadmap efficiently by exploiting tools from the Markov chain theory (Section 3). We show formally that SRS converges to the same distribution as the Monte Carlo method (Section 4). Our approach provides an efficient algorithm for computing the transmission coefficients for protein folding (Sections 5 and 6). It also has potential applications in other questions regarding the kinetics of molecular motion (Section 7).

2. Stochastic roadmap simulation

In Stochastic Roadmap Simulation, we first construct a roadmap, as a discrete representation of molecular motion. A roadmap represents a large number of possible Monte Carlo simulation paths simultaneously and enables us to perform key computation efficiently.

2.1 Conformation space

The conformation of a molecule can be specified in various ways. In a lattice model, we specify the lattice positions of constituent atoms. In protein folding, we commonly use the backbone torsional angles (ϕ and ψ) of a protein. SRS is applicable to all these different representations, provided that the conformation of a molecule can be specified as a finite number of parameters that uniquely determine the 3-D position of every atom in the molecule. Formally, a conformation of d parameters is specified by a vector $(\theta_1, \theta_2, \ldots, \theta_d)$.

The set of all possible conformations form the *conformation space* $\mathcal{C}$. A point in $\mathcal{C}$ corresponds to a particular assignment to the parameters that specify the conformation of the molecule.

The conformational parameters determine the interaction between atoms of the molecule and between the molecule and the medium, *e.g.*, the van der Waals and electrostatic forces. These interactions give rise to the attractive and repulsive forces that dictate the motion of a molecule. SRS assumes that the interactions are described by an energy function $E(q)$, which depends only on the conformation q of the molecule; it does not require E to have any particular properties or functional forms.

2.2 Roadmap construction

We encode many pathways in $\mathcal{C}$ with a directed graph G, called a roadmap. Each node of the roadmap G is a randomly sampled conformation in $\mathcal{C}$. Each (directed) edge between two nodes v_i and v_j carries a weight P_{ij}, which is the probability for the molecule to transition from v_i to v_j. The probability P_{ij} is 0 if there is no edge between v_i and v_j. Otherwise, the value of P_{ij} depends on the energy difference between v_i and v_j. We thus adopt a stochastic view of molecular motion: P_{ij} represents the probability that the molecule will next move to conformation v_j, given that it is currently in v_i.

To construct the roadmap, our algorithm samples n conformations independently at random from $\mathcal{C}$. For simplicity, we use the uniform sampling distribution by picking a value uniformly at random for each conformational parameter $\theta_i, i = 1, 2, \ldots$ from its allowable range. For every node v_i, we find the k nearest neighbors of v_i, according to a suitable metric such as the RMS or Euclidean distance in $\mathcal{C}$. Let N_i denote the set of k nearest neighbors of v_i. The algorithm then computes the transition probability P_{ij} between every pair of neighboring nodes v_i and v_j, where v_j is in N_i. P_{ij} is computed based on $\Delta E_{ij} = E(v_j) - E(v_i)$, the energy difference between the conformations v_i and v_j. In formula, we have

$$P_{ij} = \begin{cases} (1/|N_i|) \exp(-\Delta E_{ij}/k_{\mathrm{B}}T), & \text{if } \Delta E_{ij} > 0; \\ 1/|N_i|, & \text{otherwise}; \end{cases}$$

where k_{B} is the Boltzmann constant, T is the temperature, and $|N_i|$ is the number of neighbors of node v_i, excluding itself. If a node v_j is not in N_i, then v_i and v_j are too far apart for their energy difference to be a good basis for estimating the transition probability, and we set $P_{ij} = 0$. Finally we define the self-transition probabilities:

$$P_{ii} = 1 - \sum_{j \neq i} P_{ij},$$

which ensures that the transition probabilities from any node sum up to 1.

The transition probabilities thus defined are consistent with the Metropolis criterion used in Monte Carlo simulation. They allow us to establish a connection between SRS and Monte Carlo simulation formally (see Section 4). In comparison, previous work uses roadmaps with heuristic edge weights based on the energy differences [SLB99, ASBL01, SA01]. They do not have the same interpretation of the roadmap as representing the stochastic motion of molecules, and thus cannot be formally validated in the same stochastic framework that we use here. Furthermore, with our interpretation, we can exploit the knowledge from Markov chain theory to efficiently process interesting queries (see Section 3).

2.3 Using SRS to study molecular motion

Typically, Monte Carlo simulation generates random paths through $\mathcal{C}$ in search of the global minimum of the energy function E. Such paths are interesting for understanding the energy landscape and exploring the kinetics of molecular motion, as well as determining the native folds of proteins and the binding sites in ligand-protein docking (see, *e.g.*, [Fer99, KS96]).

A path generated by Monte Carlo simulation corresponds to a sequence of random moves in the conformation space $\mathcal{C}$. Such a pathway in $\mathcal{C}$ can also be obtained by following a sequence of edges in our roadmap G: at node v_i, we decide which node to move to next according to the transition probabilities P_{ij}.

With our choice of transition probabilities, there is a strong relationship between paths generated by SRS, *i.e.*, paths in the roadmap, and paths generated by Monte Carlo simulation. The main difference between SRS and Monte Carlo simulation is the space that they operate on. SRS operates on the set of sampled conformations, while Monte Carlo method operates on the underlying continuous conformation space $\mathcal{C}$. So SRS can be regarded as a discretely sampled version of Monte Carlo simulation.

In [SKS01], it is argued that Monte Carlo simulation can be applied to the understanding of protein folding kinetics. The relationship between SRS paths and Monte Carlo simulation paths suggests that their analysis is applicable to our approach as well.

However, Monte Carlo simulation focuses on only one pathway at a time and easily gets stuck in the local minima of the energy function, repeatedly sampling many similar conformations without obtaining much new information. SRS constructs a roadmap containing many Monte Carlo simulation paths by sampling directly from the space of all pathways. It processes these paths together using algebraic methods, thus greatly reducing computation time, as we will see in the next sections. The computation does not suffer from the local-minima problem encountered in Monte Carlo simulation.

3. First-step analysis and roadmap query

A roadmap G contains a multitude of information about molecular motion. Given two nodes v_i and v_j in G, we can easily compute the most likely pathway from v_i to v_j by searching for a minimum-weight path from v_i to v_j in a graph similar to G but with $-\ln P_{ij}$ as edge weights. This leads to results similar to those in the earlier work [SLB99, ASBL01, SA01], which use a directed graph with heuristic edge weights based on energy differences, because the heuristic edge weights can be interpreted as probabilities. However, an insight resulting from our choice of transition probabilities is that a roadmap implicitly defines a Markov chain that captures the stochastic nature of molecular motion. This allows us to take advantage of powerful tools from the Markov chain theory. We now focus on one such tool, the first-step analysis, which will be used later to study the kinetics of protein folding.

Consider a roadmap G representing the motion of a protein during the folding process. Let $\mathcal{F}$ be a set of nodes in G that lie in the folded state. In other words, they are structurally similar to the native fold. The set $\mathcal{F}$ is an example of a *macrostate*, which is defined as a set of roadmap nodes that share a common property. A macrostate is an abstraction that combines a set of nodes into a single entity. Now suppose that we are interested in finding, for every node v_i in G, the expected number of transitions that it takes to go from v_i to the folded state, *i.e.*, any node in $\mathcal{F}$. Denote this number by t_i. The naive approach to compute t_i would be to perform many Monte Carlo simulation runs, starting from v_i and average the number of steps taken by each run to get an estimate of t_i. This approach has a high variance in the estimate, due to the stochastic nature of the simulation procedure and thus requires a large number of simulation runs for each node v_i in order to get reasonable results. In contrast, the first-step analysis solves for all t_i simultaneously, without the need for explicit simulation.

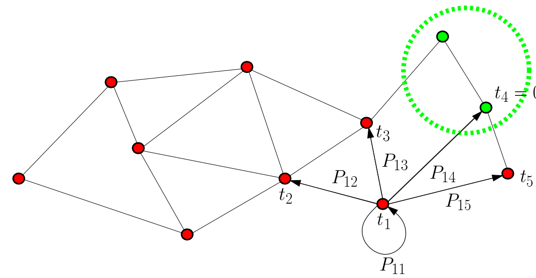

Figure 2: The first-step analysis.

The first-step analysis proceeds by conditioning on what happens after the first step. Suppose that we start at some node $v_i \notin \mathcal{F}$ and perform one step of transition. First t_i is increased by one. Then, in the next step, we reach either the folded state or another node $v_j \notin \mathcal{F}$. In the former case, we simply stop. In the latter case, the expected number of steps from then on is given by t_j, by the definition. More formally, we have the following system of self-consistent equations:

$$t_i = 1 + \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 0 + \sum_{v_j \notin \mathcal{F}} P_{ij} \cdot t_j, \quad \text{for every } v_i \notin \mathcal{F}. \quad (1)$$

In the second term of (1), P_{ij} is multiplied by zero, because we stop as soon as we reach the folded state. See Figure 2 for an illustration.

The linear system in (1) contains one equation and one unknown for each node v_i not in $\mathcal{F}$. A unique solution to (1) is guaranteed to exist, because the roadmap G contains only one strongly-connected component by construction, and so the Markov chain represented by G is ergodic [TK94]. By solving the linear system algebraically, we obtain t_i for all the nodes simultaneously, without any explicit simulation.

Since there are usually many nodes in the roadmap, the linear system is large and is solved with an iterative method. A simple one is the Jacobi iteration [PTVP92]. Let $t_i^{(k)}$ be the value of t_i at kth iteration. We set $t_i^{(0)} = 0$ for all i and repeatedly apply

$$t_i^{(k+1)} = 1 + \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 0 + \sum_{v_j \notin \mathcal{F}} P_{ij} \cdot t_j^{(k)},$$

until the iteration converges, *i.e.*, $\|t^{(k+1)} - t^{(k)}\|_\infty \leq \varepsilon$ for some small pre-selected constant ε. The convergence rate of the Jacobi method is slow, but a better method such as Gauss-Seidel or successive overrelaxation (SOR) can be used instead to improve the performance [PTVP92]. In addition, every node in G is, by construction, connected to at most k neighboring nodes, for some pre-selected value of k. Usually k is much smaller than the total number of nodes n. We can exploit this feature and run a sparse matrix ordering algorithm (see, *e.g.*, [GL89, GMS92]) to produce a linear system that has a banded, sparse structure, which can greatly improve the running time of iterative solvers.

The first-step analysis is actually more general than the simple example suggests here. For instance, we may have multiple macrostates and want to know the probability of reaching one macrostate before the others. We will discuss how to use the first-step analysis to deal with this in Section 5.

4. Formal link between SRS and Monte Carlo simulation

So far, we have presented the roadmap and the first-step analysis as an efficient way to represent molecular motion pathways and extract interesting information from them. Before applying these techniques to relevant biological questions, we would like to formalize the relationship between SRS and Monte Carlo simulation, a classic technique for studying molecular motion. We prove that SRS and Monte Carlo simulation converge to the same distribution. The proof has several implications. First, it validates our intuition that the quality of results from SRS improves as the roadmap size increases. Second, it provides the basis for analyzing the approximation error for any given roadmap size. Finally, it gives us the formal justification for using our probabilistic edge weights P_{ij}, rather than the heuristic edge weights in the earlier work using roadmaps.

4.1 Stationary distribution

To establish the proof, we now briefly describe the concept of the *stationary distribution* of a Markov chain.

Every roadmap G defines a Markov chain, which has an associated limiting distribution π obtained as follows: Perform a random walk on G, starting at a node in G. At each step of the random walk, make a move to the next node according to the transition probabilities P_{ij}. If we let the random walk continue infinitely, then under the condition that the roadmap is ergodic, the starting node becomes irrelevant: in the limit, each node v_i is visited with a fixed probability π_i according to π, regardless of the starting node. So π describes the limiting behavior of *all* possible simulated random walks on the roadmap. Since the roadmap G represents the motion of molecules, for each node v_i in G, π_i gives the fraction of the time that the molecule spends at v_i in the limit.

The limiting distribution π can be shown to satisfy the following self-consistent equations [TK94]:

$$\pi_i = \sum_j \pi_j P_{ji}, \quad \text{for all } i. \tag{2}$$

With the additional constraints $\pi_i \geq 0$ for all i and $\sum_i \pi_i = 1$, the solution to (2) is guaranteed to be a well-defined probability distribution. Equation (2) says that, in the limit, the distribution π no longer changes from one step of the random walk to the next. For this reason, π is called the *stationary distribution*. Now we are ready to relate the stationary distribution of SRS to the limiting distribution of Monte Carlo simulation.

4.2 Limit behavior of SRS.

Consider a molecule moving through the conformation space $\mathcal{C}$ governed by the energy function E. In the limit, the probability distribution of conformations visited by the molecule is given by the Boltzmann distribution β [Lea96]. Specifically the density β at a particular point v of $\mathcal{C}$ is

$$\beta(v) = \frac{1}{Z_\beta} \exp(-E(v)/k_{\mathrm{B}}T), \tag{3}$$

where $Z_\beta = \int_{\mathcal{C}} \exp(-E(v)/k_{\mathrm{B}}T)dv$ is a normalization constant, also known as a partition function.

It is well-known that the limiting distribution of Monte Carlo simulation is β, the Boltzmann distribution [Lea96]. This means that if we allow the Monte Carlo simulation to continue infinitely, the sampled conformations will distribute according to β. We would like to answer the same question for SRS. What is the limit behavior of SRS?

Note that the conformation space $\mathcal{C}$ is continuous. Thus (3) represents a probability *density* function over $\mathcal{C}$. To compute the probability for a set of conformations, we need to integrate the density function β over the set. More formally, let $\mathcal{S} \subseteq \mathcal{C}$ be any subset of the conformation space. In the limit, the probability that a conformation in $\mathcal{S}$ is sampled by a Monte Carlo simulation process is

$$\beta(\mathcal{S}) = \int_{\mathcal{S}} \beta(v)dv.$$

Now consider a roadmap with stationary distribution π given by (2). The first question is how to estimate the probability of the set $\mathcal{S}$ using the roadmap. The roadmap contains a set of discretely sampled nodes from $\mathcal{C}$. So the estimate can be obtained by the summing the stationary distribution π over all the nodes v_i that lie in the set $\mathcal{S}$. Some nodes in the roadmap may have more neighbors than others, and so we normalize the sum by $|N_i|$, the number of neighbors of node v_i. In formula, we have

$$\pi(\mathcal{S}) = \frac{1}{Z} \sum_{v_i \in \mathcal{S}} \frac{\pi(v_i)}{|N_i|},$$

where $Z = \sum_i \pi(v_i)/\,|N_i|$ is a normalizing constant.

If SRS represents the stochastic motion of molecules with the same limit behavior as Monte Carlo simulation, then the limit distributions of these two methods should converge. In other words, as the roadmap size increases, $\pi(\mathcal{S})$ should approach $\beta(\mathcal{S})$ for any subset $\mathcal{S}$ in $\mathcal{C}$. This is stated formally in Theorem 1.

THEOREM 1. *Let $\mathcal{S}$ be any subset of the conformation space $\mathcal{C}$ with relative volume $\mu(\mathcal{S}) > 0$. For any $\varepsilon > 0$, $\delta > 0$, and $\gamma > 0$, there exists N, such that in a roadmap with N uniformly sampled nodes, the difference between the probability $\beta(\mathcal{S})$ and the estimate $\pi(\mathcal{S})$ from the roadmap is given by*

$$(1-\delta)\beta(\mathcal{S}) - \varepsilon \leq \pi(\mathcal{S}) \leq (1+\delta)\beta(\mathcal{S}) + \varepsilon, \tag{4}$$

with probability at least $1-\gamma$.

Furthermore, if $\varepsilon \leq \min\{Z_\beta/2, \mu(\mathcal{S})/Z_\beta\}$ and $\delta \leq Z_\beta/(Z_\beta + 2)$, then the number of roadmap nodes N required is given by

$$N = \max\left\{ \frac{32\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S}{\mu(\mathcal{S})Z_\beta^2\varepsilon^2} + 1, \right.$$
$$\left. \frac{\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S\,(Z_\beta+2)^4}{4Z_\beta^2\mu(\mathcal{S})^3\delta^2} + 1 \right\},$$

where $\|f\|_S = \sup_v f(v) - \inf_v f(v)$.

PROOF. See Appendix A. □

The first part of the above theorem says that, with high probability, the stationary distribution π associated with a roadmap can approximate β, the Boltzmann distribution to any desired level of accuracy characterized by the relative error δ and the absolute error ε. Since Monte Carlo simulation approaches β in the limit, too, it follows from Theorem 1 that both SRS and Monte Carlo simulation converge to the same limit distribution.

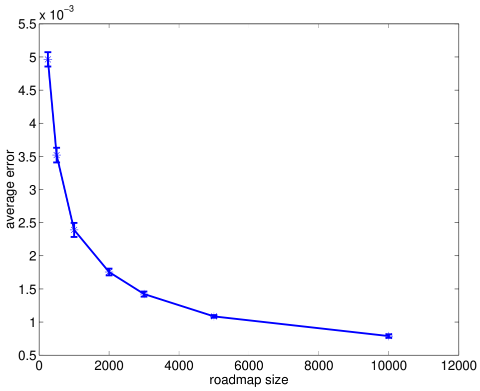

Figure 3: Error in SRS estimates of the stationary distribution.

Figure 3 illustrates the result of Theorem 1. It shows that the error in our roadmap estimates of the stationary distribution decreases with increasing roadmap sizes, as predicted by the theorem. Figure 3 was obtained by evaluating our roadmap estimates of stationary distribution on a synthetic energy landscape in a 2-D conformation space. The space is divided into 100 equally-sized bins $B_i, i = 1, 2, \ldots, 100$. We generated roadmaps of increasing sizes and computed the stationary distribution $\pi(B_i)$ on the roadmap. The Boltzmann distribution $\beta(B_i)$ for each bin B_i was estimated by Monte Carlo integration. Figure 3 shows the average error in our estimates, *i.e.*, $(1/100)\sum_{i=1}^{100} |\pi(B_i) - \beta(B_i)|$.

The second part of Theorem 1 deals with the convergence rate of the roadmap estimate. For any desired level of approximation (a given absolute error ε, relative error δ, and confidence level γ), the number of milestones required is polynomial in $1/\varepsilon$, $1/\delta$, and $\ln(1/\gamma)$. The size of the roadmap also depends polynomially on the range of values for the Boltzmann faee ctor $\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S$, the partition function Z_β, and the inverse of the relative volume of the set of interest $1/\mu(\mathcal{S})$, where the relative volume $\mu(\mathcal{S})$ is defined as the ratio of the volume of the set $\mathcal{S}$ to the total volume of $\mathcal{C}$.

Our analysis indicates that SRS and Monte Carlo simulation sample points from the same distribution in the limit. It does not, however, compare the paths generated by these two methods. In Section 2.3, we explained intuitively the connection between the paths in SRS and those in Monte Carlo simulation. Although this connection has not been proven formally, it seems clear that, as the number of milestones increases, a path in the roadmap will correspond to a path in Monte Carlo simulation with infinitesimal step sizes in the limit.

We have described, in the foregoing sections, how to construct a roadmap and perform queries using the first-step analysis, and have shown the relationship between SRS and Monte Carlo Simulation. Now we are ready to apply SRS to interesting biological questions.

5. Computing the transmission coefficients for protein folding

Protein folding is the one of the most marvellous processes in nature. Under suitable conditions, most proteins go through a series of geometric transformations and assume unique 3-D structures, called the native folds, which allow them to perform intricate biological functions. Since the pioneering work of Anfinsen [Anf73], there has been large, on-going effort on predicting the native structure of a protein, given its amino acid sequence (see [KS96] for a survey). Equally interesting, however, is to understand the folding process itself: What geometric transformations does the protein go through during the folding? Which conformations are "closer" to the native structure along the folding pathway?

To address this kind of questions, the transmission coefficient has been introduced as an order parameter for protein folding [DPG$^+$98]. It gives an indication on how far away a conformation is from the native structure kinetically. For a folding process dominated by two stable macrostates, a folded state $\mathcal{F}$ and an unfolded $\mathcal{U}$, the transmission coefficient τ for a given conformation q is the probability of arriving in $\mathcal{F}$ before arriving in $\mathcal{U}$, starting from q. If q is in $\mathcal{U}$, then $\tau = 0$. If q is in $\mathcal{F}$, then $\tau = 1$. More importantly, the transmission coefficient measures the "kinetic distance" between a given conformation and the folded state (or the unfolded state): from any conformation q with $\tau > 0.5$, the protein is more likely to fold first than to unfold first, and therefore q is closer to the folded state [DPG$^+$98]. The transmission coefficient is not associated with any particular folding pathway, but depends on many pathways from one state to another. It thus describes the average behavior of a folding process from a given conformation.

Using SRS, we can easily compute transmission coefficients. Let $v_i, i = 1, 2, \ldots$ be the sampled nodes in the roadmap, and τ_i be the transmission coefficient of v_i. After constructing the roadmap, we apply the first-step analysis to establish the following relationship for every node v_i not in $\mathcal{F}$ or $\mathcal{U}$:

$$\tau_i = \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 1 + \sum_{v_j \in \mathcal{U}} P_{ij} \cdot 0 + \sum_{j \notin \mathcal{F}, j \notin \mathcal{U}} P_{ij} \cdot \tau_j. \tag{5}$$

Equation (5) is obtained by conditioning on the first transition. After one step of transition, we have three possibilities. In the first case, we reach a node in $\mathcal{F}$, the folded state. So we have reached $\mathcal{F}$ before $\mathcal{U}$ with probability 1. In the second case, we reach a node in $\mathcal{U}$. So we have reached $\mathcal{U}$ before $\mathcal{F}$, and the probability of reaching $\mathcal{F}$ before $\mathcal{U}$ is 0. Finally, if we reach a node v_j in neither $\mathcal{F}$ or $\mathcal{U}$, then the probability depends on the value of τ_j. Again, the linear system in (5) can be solved iteratively to obtain the transmission coefficients for all the nodes in the roadmap simultaneously without explicit simulation.

6. Results

We now show the computed results on two examples. The first one is based on a relatively simple synthetic energy landscape, and the second one, on a real protein. We compare the results from SRS with those obtained from Monte Carlo simulation, a standard method for computing transmission coefficients, and demonstrate that SRS reduces the running time by several orders of magnitude and is more accurate. A main reason for us to use the synthetic data is that Monte Carlo simulation would take excessive amount of computation time on a fast workstation if a real protein were used, and so we cannot perform extensive comparison. Our preliminary implementation of SRS solves linear systems (5) with the Jacobi method and does not exploit the sparsity of linear systems. The computational advantage of SRS could be further increased by using a better linear system solver.

The synthetic energy landscape lies in a 2-D conformation space. It is constructed using a linear combination of radially symmetric Gaussians. The centers, the decay rates, and the heights of the Gaussians are chosen at random. There is also a paraboloid centered at the origin. The landscape has two global minima.

In the first test, we used SRS to compute τ for 101 sampled nodes in the conformation space, with a roadmap of 10102 nodes, and then used Monte Carlo simulation to compute the results for the same

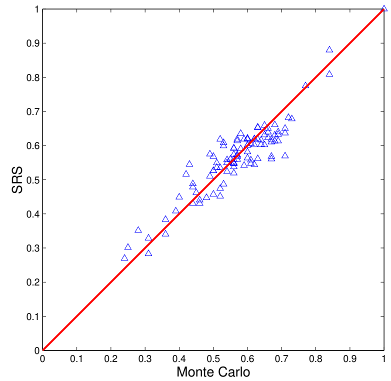

Figure 4: The correlation of transmission coefficients computed by Monte Carlo simulation and SRS on synthetic data.

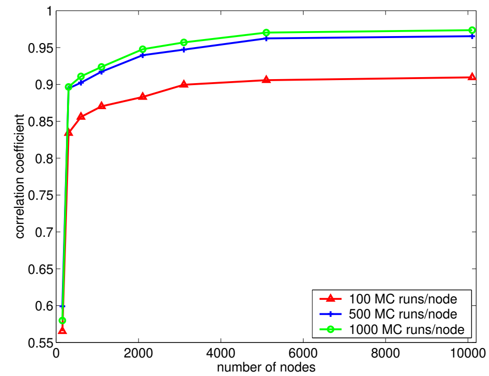

Figure 5: The change of correlation coefficient ρ as the number of nodes in the roadmap increases. The three curves correspond to Monte Carlo simulation with $r = 100, 500, 1000$ independent runs for each node. As r increases, the correlation between the results from the two methods usually improves.

nodes. In the Monte Carlo simulation, we performed 500 independent runs for each node. In each step of the simulation, we propose a new conformation q' in a neighborhood around the current conformation q and accept or reject q' according to the Metropolis criterion. Each simulation run stops as soon as it is within a small neighborhood of a conformation in the folded or the unfolded state. The results computed with the two methods are plotted along the horizontal and vertical axes respectively (Figure 4). The figure shows that all the points lie close to the diagonal line, indicating that the results from the two different methods are in good correspondence.

To examine the accuracy of our computed results, we conducted further tests by varying the number of nodes sampled by SRS and the number of independent Monte Carlo simulation runs for each node. In each test, we summarize the correspondence between the results from the two methods by their (normalized) correlation coefficient, which is defined as

$$\rho(x, y) = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sqrt{(\langle x^2 \rangle - \langle x \rangle^2)(\langle y^2 \rangle - \langle y \rangle^2)}}$$

for two vectors x and y, where $\langle \cdot \rangle$ denotes the operation of taking the expectation. Note that the magnitude of ρ is always between 0 and 1, with 0 indicating no correlation and 1 indicating perfect correlation. Figure 5 shows the results of these additional tests. The horizontal axis of the graph is the number of nodes in the roadmap, and the vertical axis is the correlation coefficient ρ. The graph contains three curves, each corresponding to a different number of independent Monte Carlo simulation runs per node. The three curves show a generally similar trend. Initially ρ improves rather quickly as the number of nodes in the roadmap increases. The curves then flatten out after a certain point. It is not immediately clear whether they will reach 1, which would indicate perfect correlation. Since ρ measures only the relationship between the two methods, and we do not know the ground truth, these general trends do not tell us whether the discrepancy is due to the inaccuracy in SRS or the variance inherent in Monte Carlo simulation. However, we can get a hint by comparing the three curves. For a roadmap of a given size, ρ generally improves as we increase the number of independent Monte Carlo simulation runs. This seems to indicate that SRS gives the more accurate results: when we increase the number of independent Monte Carlo simulation runs per node, the variance of Monte Carlo simulation decreases, and the results get closer to those obtained from SRS.

We also compared the running time of the two methods. The computation time of SRS consists of two parts: the time to construct the roadmap and the time to solve a linear system of equations. On a real protein, the first part is dominated by the time to evaluate the energy of sampled nodes. The second part depends on the size of the linear system, which, in turn, depends on the number of nodes in the roadmap. The running time of Monte Carlo simulation is dominated by the time to compute the energy of sampled conformations.

Table 1: Running times of 100 Monte Carlo simulation runs per conformation on the synthetic energy landscape. First and third rows give the number of conformations processed. Second and fourth give the running times.

No. Conf.	10	20	30	40	50
Time (sec.)	866	1588	2356	3191	4026
No. Conf.	60	70	80	90	100
Time (sec.)	4913	5621	6404	7203	8077

In our current implementation, the roadmap construction part of SRS was coded in C++, and the linear system solver, in Matlab. Monte Carlo simulation was implemented entirely in C++. The timing results reported here were obtained on a 1GHz Pentium-III PC with 1GB of memory. In a typical run on the synthetic landscape, SRS took 8 seconds to construct a roadmap of about 10,000 nodes, and 750 seconds to solve the linear system and obtain the transmission coefficients for all the nodes. The running times of Monte Carlo simulation is tabulated in Table 1. It is clear from Table 1 that the running time of Monte Carlo simulation is linear with respect to the number of conformations processed. Although we did not try the Monte Carlo simulation on all 10,000 conformations, it is not difficult to infer that the running time would be around 800,000 seconds.

In addition to the synthetic data, we also tested our algorithm on a real protein, repressor of primer (ROP). ROP is a four-helix bundle. We study one monomer in isolation as in [STD95].

The 3-D structure of ROP is obtained from the Protein Data Bank [B+77]. The monomer consists of 56 residues forming two α helices connected by a loop. Our implementation specifies the conformation of the monomer with a vector-based representation [SB97, ASBL01]. The protein is represented by two vectors connected by a loop. As in [ASBL01], there are six conformational parameters in total. Our energy function uses the H-P model [STD95] consist-

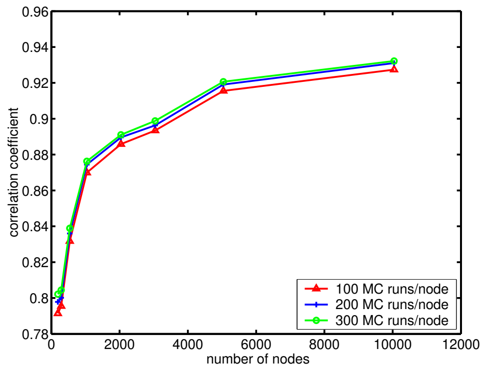

Figure 6: The correlation of the transmission coefficients of ROP computed by SRS and Monte Carlo simulation.

ing of two terms: one measuring the hydrophobic interaction and the other measuring the excluded volume. In both SRS and Monte Carlo simulation, we discard conformations that violate steric hindrance constraints.

Here, the folded macrostate contains all conformations within 3 Å of the native fold according to the RMS distance, and the unfolded macrostate, all the conformations within 10 Å of the fully-extended conformation.

We examined the transmission coefficients for ROP at 39 randomly selected conformations using both SRS and Monte Carlo simulation. In SRS, we computed the estimates for increasing roadmap sizes. In Monte Carlo simulation, we performed up to 300 independent runs at each of the 39 conformations to get the estimates. As in the case for synthetic data, we computed the correlation coefficient for increasing number of Monte Carlo simulation runs per node. The results, shown in Figure 6, suggest conclusions similar to those obtained from synthetic data. First, the SRS estimates improve rapidly as the roadmap size increases. Second, the correlation tends to increase as we perform more Monte Carlo simulation runs per node, indicating that variance in Monte Carlo simulation is the likely cause for the discrepancy.

The total time to generate a roadmap with 5000 nodes and compute the transmission coefficients for all the nodes in the roadmap was about one hour. In comparison, it took an average of *three days* of computation time on the same machine in order to execute 300 Monte Carlo simulation runs required to estimate the transmission coefficient at just *one* conformation. To generate the results for Figure 6, the Monte Carlo simulation procedure spent about one hundred days of computation time, as opposed to one hour needed by SRS for all 5000 conformations.

The above results indicate that, to compute the transmission coefficients τ for only a few nodes (say 10), we may use Monte Carlo simulation. SRS is not applicable in this case, because the number of nodes is too small to generate accurate results. On the hand, if we want τ for many nodes spread over the entire energy landscape, which is the more typical situation, SRS is far more efficient. It computes τ for all the nodes simultaneously, thus offering a speedup of several orders of magnitude in running time according to our experiences.

7. Discussion

We have introduced stochastic roadmap simulation as a new framework for studying molecular motion. A roadmap compactly encodes an ensemble of pathways, and the first-step analysis allows us to efficiently extract from the roadmap interesting kinetic properties of molecular motion. A salient feature of SRS is that it examines all the paths in a roadmap together using algebraic methods rather considering them one at a time as classic methods such as Monte Carlo simulation would do. SRS also avoids the local-minima problem encountered with existing methods. It thus gains tremendous computational efficiency for suitable problems.

SRS is closely related to Monte Carlo simulation. Every path in a roadmap can be interpreted as a possible Monte Carlo simulation run. In addition to that, we proved that, in the limit, SRS and Monte Carlo simulation converge to the same distribution.

We have applied SRS to the computation of the transmission coefficients for protein folding in order to test its effectiveness. Transmission coefficients can be obtained from Monte Carlo simulation, but the long computation time required is pointed out as one of the main obstacles to its general utility [DPG+98]. We have shown in our computational studies that SRS reduces the running time by several orders of magnitude, compared with Monte Carlo simulation, and obtains more accurate results.

In [DPG+98], Du *et al.* suggest that the transmission coefficient can serve as the best possible measure of kinetic distance for a system. However, overwhelmed by the computational burden of standard simulation methods, they wrote, "To conclude, we stress that we do not suggest using the transmission coefficient as a transition coordinate for practical purposes as it is very computationally intensive." Our computational studies suggest that SRS makes the computation of the transmission coefficient viable, which would potentially enable its use in practice.

Our preliminary work indicates that SRS is a promising new method for studying molecular motion, but many interesting questions remain to be explored.

We would like to find better ways for constructing roadmaps. Currently we use the uniform distribution to sample the conformation space $\mathcal{C}$ of a molecule. As the dimension of $\mathcal{C}$ gets higher, it becomes increasingly more difficult to obtain biologically interesting conformations with uniform sampling. There are two ways to deal with this issue. First, we may use a more efficient representation (*e.g.*, the vector-based representation in Section 6) to reduce the dimension of $\mathcal{C}$. Second, we may construct better sampling strategies that favor low-energy regions in $\mathcal{C}$. If the nodes of a roadmap are sampled non-uniformly, we must make appropriate adjustment when assigning transition probabilities so that SRS converges to the same distribution as Monte Carlo simulation does.

The most time-consuming part of the first-step analysis is solving a linear system of equations. The linear system solver in our current implementation is still crude. A better iterative method that exploits the sparse structure of a linear system will certainly further improve the efficiency of SRS.

Even more interesting is the application of SRS to other important questions on molecular motion, such as the order of events in protein folding. Do secondary structure elements (SSE) form first as sub-units before they organize into a tertiary structure? When does a particular SSE appear? We also plan to use SRS for studying the kinetics of ligand-protein binding. Our hypothesis is that there are energy barriers around binding conformations. If a ligand is in a binding conformation, it must overcome the energy barrier in order to escape. The conformation that puts the ligand at the catalytic site has the highest energy barrier, and it takes the longest time to escape from it. The roadmap combined with the first-step analysis offers a natural tool to compute the escape times and thus identify potential catalytic sites. We believe that SRS will play an important role in studying these and other interesting biological questions.

Acknowledgments: This work has been partially funded by an NSF-ITR grant and a grant from Stanford's Bio-X program. Apaydin was supported by the D.L. Cheriton Stanford Graduate Fellowship. Brutlag was supported National Human Genome Research Institute grant HGF02235. Guestrin was supported by a Siebel Scholarship and by the Sloan Foundation. This paper has greatly benefitted from discussions with D. Koller, V. Pande, A. Singh, and J. Snoeyink.

References

[Anf73] C.B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.

[ASBL01] M.S. Apaydin, A.P. Singh, D.L. Brutlag, and J.C. Latombe. Capturing molecular energy landscapes with probabilistic conformational roadmaps. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2001.

[B+77] F.C. Bernstein et al. The protein data bank: A computer-based archival file for macromolecular structure. *Journal of Molecular Biology*, 112(3):535–542, 1977.

[BOSW95] J.D. Bryngelson, J.N. Onuchic, N.D. Socci, and P.G. Wolynes. Funnels, pathways, and the energy landscape of protein folding: A synthesis. *Proteins: Structure, Function, and Genetics*, 21(3):167–195, 1995.

[DC97] K.A. Dill and H.S. Chan. From levinthal to pathways to funnels. *Nature Structural Biology*, 4(1):10–19, 1997.

[DK99] C.M. Dobson and M. Karplus. The fundamentals of protein folding: Bringing together theory and experiment. *Current Opinion in Structural Biology*, 9:92–101, 1999.

[DPG+98] R. Du, V. Pande, A.Y. Grosberg, T. Tanaka, and E. Shakhnovich. On the transition coordinate for protein folding. *Journal of Chemical Physics*, 108(1):334–350, 1998.

[Fer99] A. Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. W.H. Freeman & Company, New York, 1999.

[GL89] A. George and J. Liu. The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31(1):1–19, 1989.

[GMS92] J.R. Gilbertand, C. Moler, and R. Schreiber. Sparse matrices in matlab: Design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356, 1992.

[Hai92] J.M. Haile. *Molecular Dynamics Simulation: Elementary Methods*. John Wiley & Sons, New York, 1992.

[Haj88] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2):311 – 329, 1988.

[Hoe63] W. Hoeffding. Probability inequalities for sum of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[KS96] A. Kolinski and J. Skolnick. *Lattice Models of Protein Folding, Dynamics and Thermodynamics*. Chapmann & Hall, New York, 1996.

[KŠLO96] L. Kavraki, P. Švestka, J.C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[KW86] M.H. Kalos and P.A. Whitlock. *Monte Carlo Methods*, volume 1. John Wiley & Son, New York, 1986.

[Lea96] A.R. Leach. *Molecular Modelling: Principles and Applications*. Longman, Essex, England, 1996.

[PGTR98] V.S. Pande, A.Y. Grosberg, T. Tanaka, and D.S. Rokhsar. Pathways for protein folding: Is a new view needed? *Current Opinion in Structural Biology*, 8:68–79, 1998.

[PTVP92] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Plannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

[SA01] G. Song and N.M. Amato. Using motion planning to study protein folding pathways. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 287–296, 2001.

[SB97] A.P. Singh and D.L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, pages 284–293, 1997.

[SKS01] J. Shimada, E.L. Kussell, and E.I. Shakhnovich. The folding thermodynamics and kinetics of crambin using an all-atom monte carlo simulation. *Journal of Molecular Biology*, 308(1):79–95, 2001.

[SLB99] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexible ligand binding. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, pages 252–261, 1999.

[STD95] S. Sun, P.D. Thomas, and K.A. Dill. A simple protein folding algorithm using a binary code and secondary structure constraints. *Protein Engineering*, 8:769–778, 1995.

[TK94] H. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, New York, 3rd edition, 1994.

Appendix

A. Proof of Theorem 1

In the first set of the proof, we show a closed form solution to the stationary distribution on the roadmap:

LEMMA 2. *For any set $\mathcal{S}$, the stationary distribution on the roadmap can be rewritten as:*

$$\pi(\mathcal{S}) = \frac{\sum_{i \in \mathcal{S}} \exp(-E(v_i)/k_\mathrm{B}T)}{\sum_j \exp(-E(v_j)/k_\mathrm{B}T)}.$$

PROOF. First, note that a random walk on a graph with our transition probabilities P_{ij} has a stationary distribution:

$$\pi_i = \pi_G(v_i) \exp(-E(v_i)/k_\mathrm{B}T)/Z_U; \tag{6}$$

where Z_U is a normalization constant and $\pi_G(v_i)$ is the stationary distribution for a random walk on a graph with the same connectivity, but with the energy dependent $\exp(-\Delta E_{ij}/k_\mathrm{B}T)$ term removed from the transition probabilities P_{ij} (for a derivation of this statement, see for example [Haj88]). We will denote these new transition probabilities by P'_{ij}.

Now note that for the case of our roadmaps, P'_{ij} becomes an uniform distribution over the neighbors:

$$P'_{ij} = \begin{cases} \frac{1}{|N_i|} & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases} ;$$

where N_i is the set of neighbors of node i; and $P'_{ii} = 0$. Note that in our roadmaps node i is connected to j ($j \in N_i$) if and only if node j is connected to i ($i \in N_j$). In such graph, the stationary distribution π_G is trivially given by:

$$\pi_G(v_i) = \frac{|N_i|}{Z_G}; \tag{7}$$

where Z_G is a normalization constant.

Now, returning to our definition of stationary distribution in a set $\mathcal{S}$ from Section 4:

$$\pi(\mathcal{S}) = \frac{1}{Z} \sum_{i \in \mathcal{S}} \frac{\pi_i}{|N_i|}.$$

Substituting in Equations (6) and (7):

$$\pi(\mathcal{S}) = \frac{1}{Z'} \sum_{i \in \mathcal{S}} \exp(-E(v_i)/k_{\mathrm{B}}T).$$

We can compute the normalization constant Z' by the constraint $\pi(\mathcal{C}) = 1$, yielding:

$$Z' = \sum_j \exp(-E(v_j)/k_{\mathrm{B}}T).$$

Thus, proving the lemma. $\square$

The remainder of our proof will require the application of Hoeffding's inequality. We present here the simplified version of the inequality needed for the proof:

LEMMA 3 (HOEFFDING'S INEQUALITY [HOE63]). *Let Y be a random variable distributed according to $P(Y)$ such that $Y \in [a, b]$. Let $Y_1, \ldots, Y_n$ be n independent, identically distributed samples from $P(Y)$ and the empirical mean $\overline{Y} = \frac{1}{n}\sum_i Y_i$, then:*

$$\begin{aligned} &P(\overline{Y} - E[Y] \geq \varepsilon) \leq e^{-\frac{n\varepsilon^2}{b-a}}, \quad \textit{and} \\ &P(E[Y] - \overline{Y} \geq \varepsilon) \leq e^{-\frac{n\varepsilon^2}{b-a}}. \quad \square \end{aligned} \tag{8}$$

For simplicity of presentation, assume without loss of generality that the volume of the conformation space is one: $\mu(\mathcal{C}) = 1$, where the volume of some set $\mathcal{F}$ is denoted by $\mu(\mathcal{F})$, i.e., $\mu(\mathcal{F})$ represents the proportion of the total volume of $\mathcal{C}$ occupied by $\mathcal{F}$.

Theorem 1 holds for *any* confidence level $\gamma > 0$. In the proof, we will divide this γ in three parts: $\gamma_1 > 0$, $\gamma_2 > 0$ and $\gamma_3 > 0$, such that $\gamma_1 + \gamma_2 + \gamma_3 \leq \gamma$ as our proof will require three applications of Hoeffding's inequality.

Our first lemma will bound the number of points that fall in the set of interest $\mathcal{S}$:

LEMMA 4. *For a uniformly sampled roadmap of N points, for any $\varepsilon_1 > 0$, let K be the number of roadmap points that fall in the set $\mathcal{S}$, then:*

$$\mu(\mathcal{S}) - \varepsilon_1 \leq \frac{K}{N} \leq \mu(\mathcal{S}) + \varepsilon_1; \tag{9}$$

with probability at least $1 - \gamma_1$, where $\gamma_1 \geq 2e^{-N\varepsilon_1^2}$.

PROOF. Application of Hoeffding's inequality, where the random variable Y is the indicator that a point falls in the set $\mathcal{S}$. By the law of large numbers, $E[Y] = \mu(\mathcal{S})/\mu(\mathcal{C}) = \mu(\mathcal{S})$. The empirical mean $\overline{Y} = K/N$ and Y is an indicator, thus, $Y \in [0, 1]$. The proof is concluded by applying Lemma 3. $\square$

We would like to have, with high probability, at least one milestone in the $\mathcal{S}$. (This constraint can be relaxed, but the proof becomes more complicated.) Thus, we must choose the number of nodes N such that $K > 0$ with probability at least $1 - \gamma_1$. Using the constraint in Lemma 4, we know that $K \geq \lfloor N(\mu(\mathcal{S}) - \varepsilon_1) \rfloor$. Thus:

$$N \geq \lceil 1/(\mu(\mathcal{S}) - \varepsilon_1) \rceil .$$

For the remainder of the proof, we can assume, with probability at least $1 - \gamma_1$, that $K > 0$.

For the next step of the proof, we will need a definition: for some set $\mathcal{F} \subset \mathcal{C}$, let's define the *Boltzmann integral* in this set as:

$$\alpha(\mathcal{F}) = \int_{\mathcal{F}} \exp(-E(v)/k_{\mathrm{B}}T)dv.$$

Note that $\alpha(\mathcal{C})$ corresponds to the partition function Z_β. Under this definition, we can write the Boltzmann distribution as:

$$\beta(\mathcal{F}) = \frac{\alpha(\mathcal{F})}{\alpha(\mathcal{C})}.$$

We will denote the range of a function f as $\|f\|_S = \sup_v f(v) - \inf_v f(v)$. Our next lemma implies that we can estimate the Boltzmann integral with samples:

LEMMA 5. *For any set $\mathcal{F}$, let Y_i be M uniformly sampled points in $\mathcal{F}$, for any $\varepsilon > 0$, then:*

$$\alpha(\mathcal{F}) - \varepsilon \cdot \mu(\mathcal{F}) \leq \frac{\mu(\mathcal{F})}{M} \sum_i \exp(-E(Y_i)/k_{\mathrm{B}}T) \leq \alpha(\mathcal{F}) + \varepsilon \cdot \mu(\mathcal{F}); \tag{10}$$

with probability at least $1 - \gamma$, where

$$\gamma \geq 2 \exp\left(\frac{-M\varepsilon^2}{\| \exp(-E(v)/k_{\mathrm{B}}T) \|_S} \right).$$

PROOF. Define a random variable $Y = \exp(-E(v)/k_{\mathrm{B}}T)$, note that $E[Y] = \alpha(\mathcal{F})/\mu(\mathcal{F})$. The proof is concluded by applying Hoeffding's inequality. $\square$

We will apply Lemma 5 twice, first for computing the Boltzmann integral in the set $\mathcal{S}$, obtaining the bound:

$$\alpha(\mathcal{S}) - \varepsilon_2 \mu(\mathcal{S}) \leq \frac{\mu(\mathcal{S})}{K} \sum_{i \in \mathcal{S}} \exp(-E(Y_i)/k_{\mathrm{B}}T) \leq \alpha(\mathcal{S}) + \varepsilon_2 \mu(\mathcal{S}); \tag{11}$$

with probability at least: $1 - \gamma_2$, where

$$\gamma_2 \geq 2 \exp\left(\frac{-K\varepsilon_2^2}{\| \exp(-E(v)/k_{\mathrm{B}}T) \|_S} \right).$$

The second bound concerns the integral over the whole space:

$$\alpha(\mathcal{C}) - \varepsilon_3 \leq \frac{1}{N} \sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T) \leq \alpha(\mathcal{C}) + \varepsilon_3; \tag{12}$$

with probability at least: $1 - \gamma_3$, where

$$\gamma_3 \geq 2 \exp\left(\frac{-N\varepsilon_3^2}{\| \exp(-E(v)/k_{\mathrm{B}}T) \|_S} \right).$$

In the remainder of this proof, we will assume that equations (9), (11) and (12) hold, i.e., the argument holds with probability at least $1 - (\gamma_1 + \gamma_2 + \gamma_3) \geq 1 - \gamma$.

Next, note that from Lemma 2 the stationary distribution on the roadmap can be rewritten as:

$$\pi(\mathcal{S}) = \frac{\sum_{i\in\mathcal{S}} \exp(-E(Y_i)/k_{\mathrm{B}}T)}{\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)}.$$

Applying the bound on Equation (9) we get:

$$\left(\frac{\mu(\mathcal{S})-\varepsilon_1}{K/N}\right)\frac{\sum_{i\in\mathcal{S}} \exp(-E(Y_i)/k_{\mathrm{B}}T)}{\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)} \le \pi(\mathcal{S}) \le \left(\frac{\mu(\mathcal{S})+\varepsilon_1}{K/N}\right)\frac{\sum_{i\in\mathcal{S}} \exp(-E(Y_i)/k_{\mathrm{B}}T)}{\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)};$$

rearranging:

$$\left(\frac{\mu(\mathcal{S})-\varepsilon_1}{\mu(\mathcal{S})}\right)\frac{\mu(\mathcal{S})/K\sum_{i\in\mathcal{S}} \exp(-E(Y_i)/k_{\mathrm{B}}T)}{1/N\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)} \le \pi(\mathcal{S}) \le \left(\frac{\mu(\mathcal{S})+\varepsilon_1}{\mu(\mathcal{S})}\right)\frac{\mu(\mathcal{S})/K\sum_{i\in\mathcal{S}} \exp(-E(Y_i)/k_{\mathrm{B}}T)}{1/N\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)}.$$

We can now apply the bounds in Equations (11) and (12):

$$\left(\frac{\mu(\mathcal{S})-\varepsilon_1}{\mu(\mathcal{S})}\right)\frac{\alpha(\mathcal{S})-\varepsilon_2\mu(\mathcal{S})\mu(\mathcal{S})}{\alpha(\mathcal{C})+\varepsilon_3} \le \pi(\mathcal{S}) \le \left(\frac{\mu(\mathcal{S})+\varepsilon_1}{\mu(\mathcal{S})}\right)\frac{\alpha(\mathcal{S})+\varepsilon_2\mu(\mathcal{S})}{\alpha(\mathcal{C})-\varepsilon_3}.$$

This expression can be rewritten as:

$$(1-\delta)\,\frac{\alpha(\mathcal{S})}{\alpha(\mathcal{C})}-\varepsilon \quad\le\quad \pi(\mathcal{S}) \le (1+\delta)\,\frac{\alpha(\mathcal{S})}{\alpha(\mathcal{C})}+\varepsilon;$$

which finally leads us to the statement of our theorem:

$$(1-\delta)\beta(\mathcal{S})-\varepsilon \le \pi(\mathcal{S}) \le (1+\delta)\beta(\mathcal{S})+\varepsilon;$$

where ε and δ impose the following constraints:

$$\varepsilon \ \ge\ \frac{\varepsilon_2(\mu(\mathcal{S})+\varepsilon_1)}{\alpha(\mathcal{C})-\varepsilon_3}; \tag{13}$$

$$\delta \ \ge\ \frac{\varepsilon_1\alpha(\mathcal{C})+\varepsilon_3\mu(\mathcal{S})}{\mu(\mathcal{S})\,(\alpha(\mathcal{C})-\varepsilon_3)}. \tag{14}$$

In addition to these two constraints, we have the constraints imposed by the confidence levels γ_1, γ_2 and γ_3:

$$N \ \ge\ \frac{\ln(2/\gamma_1)}{\varepsilon_1^2}; \tag{15}$$

$$N \ \ge\ \frac{\ln(2/\gamma_2)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S}{(\mu(\mathcal{S})-\varepsilon_1)\varepsilon_2^2}+1; \tag{16}$$

$$N \ \ge\ \frac{\ln(2/\gamma_3)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S}{\varepsilon_3^2}; \tag{17}$$

$$\gamma \ \ge\ \gamma_1+\gamma_2+\gamma_3. \tag{18}$$

Given any $\varepsilon > 0$, $\delta > 0$ and $\gamma > 0$, we can use constraints (13) — (18) to obtain the required number of nodes N in the roadmap to satisfy the theorem.

To obtain a simpler convergence rate, we can simplify these constraints by imposing: $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \tilde{\varepsilon} \le \varepsilon$ and $\gamma_1 = \gamma_2 = \gamma_3 = \gamma/3$.

Let's first consider the ε constraint on Equation (13), which can now be written as:

$$\varepsilon \ge \frac{\tilde{\varepsilon}(\mu(\mathcal{S})+\tilde{\varepsilon})}{\alpha(\mathcal{C})-\tilde{\varepsilon}}.$$

Rearranging, we have that:

$$\tilde{\varepsilon} \le \frac{\varepsilon\alpha(\mathcal{C})}{\mu(\mathcal{S})+\varepsilon}-\frac{\tilde{\varepsilon}^2}{\mu(\mathcal{S})+\varepsilon}.$$

Finally, recall that $\mu(\mathcal{S}) \le 1$, $\tilde{\varepsilon} \le \varepsilon$ and, for a non-trivial approximation schema, $\varepsilon \le 1$. Thus, we conclude that:

$$\tilde{\varepsilon} \le \frac{\varepsilon(\alpha(\mathcal{C})-\varepsilon)}{2}. \tag{19}$$

Using a similar manipulation of the δ constraint on Equation (14), we can write:

$$\tilde{\varepsilon} \le \frac{\alpha(\mathcal{C})\mu(\mathcal{S})\delta}{\alpha(\mathcal{C})+\mu(\mathcal{S})(\delta+1)}.$$

Noting that $\mu(\mathcal{S}) \le 1$ and, in a non-trivial approximation schema, $\delta \le 1$, we can simplify the constraint as:

$$\tilde{\varepsilon} \le \frac{\alpha(\mathcal{C})\mu(\mathcal{S})\delta}{\alpha(\mathcal{C})+2}. \tag{20}$$

We can now consider the constraints on N given by Equations (15) — (17). Note that for the case of $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \tilde{\varepsilon}$ and $\gamma_1 = \gamma_2 = \gamma_3$, only the constraint in Equation (16) will be binding. This constraint can now be written as:

$$N \ge \frac{\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S}{(\mu(\mathcal{S})-\tilde{\varepsilon})\tilde{\varepsilon}^2}+1.$$

Substituting the constraints on $\tilde{\varepsilon}$ given by Equations (19) and (20), we can obtain the value of N:

$$N = \max\left\{ \frac{8\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S}{[2\mu(\mathcal{S})-\varepsilon(\alpha(\mathcal{C})-\varepsilon)]\,\varepsilon^2(\alpha(\mathcal{C})-\varepsilon)^2}+1,\ \frac{\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S}{\left[1-\left(\frac{\alpha(\mathcal{C})}{\alpha(\mathcal{C})+2}\right)\delta\right]\mu(\mathcal{S})^3\left(\frac{\alpha(\mathcal{C})}{\alpha(\mathcal{C})+2}\right)^2\delta^2}+1\right\}.$$

We can further simplify this equations by making some assumptions on ε and δ. Assuming that:

$$\varepsilon \le \frac{\alpha(\mathcal{C})}{2}; \qquad \varepsilon \le \frac{\mu(\mathcal{S})}{\alpha(\mathcal{C})}; \qquad \delta \le \frac{\alpha(\mathcal{C})}{\alpha(\mathcal{C})+2}.$$

Applying these constraints to the denominator of the equation for N above and simplifying, we obtain a final bound:

$$N = \max\left\{ \frac{32\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S}{\mu(\mathcal{S})\alpha(\mathcal{C})^2\varepsilon^2}+1,\ \frac{\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S\,(\alpha(\mathcal{C})+2)^4}{4\alpha(\mathcal{C})^2\mu(\mathcal{S})^3\delta^2}+1\right\}.$$