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

Mehmet Serkan Apaydın[*] Douglas L. Brutlag[†] Carlos Guestrin[‡]

David Hsu[§] Jean-Claude Latombe[¶] Chris Varma[||]

Abstract

Classic molecular motion simulation techniques, such as Monte Carlo (MC) simulation, generate motion pathways one at a time and spend most of their time in the local minima of the energy landscape defined over a molecular conformation space. Their high computational cost prevents them from being used to compute ensemble properties; properties requiring the analysis of many pathways. This paper introduces Stochastic Roadmap Simulation (SRS) as a new computational approach for exploring the kinetics of molecular motion by simultaneously examining multiple pathways. These pathways are compactly encoded in a graph, which is constructed by sampling a molecular conformation space at random. This computation, which does not trace any particular pathway explicitly, circumvents the local-minima problem. Each edge in the graph represents a potential transition of the molecule and is associated with a probability indicating the likelihood of this transition. By viewing the graph as a

[*]Department of Electrical Engineering, Stanford University, Stanford CA 94305

[†]Department of Biochemistry, Stanford University, Stanford CA 94305

[‡]Department of Computer Science, Stanford University, Stanford CA 94305

[§]Department of Computer Science, National University of Singapore, Singapore

[¶]corresponding author. e-mail : latombe@cs.stanford.edu Address: Department of Computer Science, Stanford University, Stanford CA 94305 Phone: (650) 723-0350 Fax: (650) 725-1449

[||]Department of EECS, MIT and Department of HST, Harvard Medical School, Cambridge, MA 02138

Markov chain, ensemble properties can be efficiently computed over the entire molecular energy landscape. Furthermore, SRS converges to the same distribution as MC simulation. SRS is applied to two biological problems: computing the probability of folding, an important order parameter that measures the "kinetic distance" of a protein's conformation from its native state; and estimating the expected time to escape from a ligand-protein binding site. Comparison with MC simulations on protein folding shows that SRS produces arguably more accurate results, while reducing computation time by several orders of magnitude. Computational studies on ligand-protein binding also demonstrate SRS as a promising approach to study ligand-protein interactions.

Key words: Monte Carlo simulation, protein folding, ligand-protein binding, probability of folding (P_{fold}), computational mutagenesis.

1 Introduction

Many essential biological processes – *e.g.*, protein folding and ligand-protein binding – depend on the ability of molecules to move and adopt different shapes over time under the influence of potential energy fields. Computational techniques play an increasing role in the analysis and understanding of such motion. In particular, Monte Carlo [KW86] and molecular dynamics [Hai92] methods are classic techniques for simulating molecular motion. But they have two major drawbacks:

- They compute individual pathways, one at a time; however, many interesting properties of molecular motion, in particular, the *ensemble properties*, are best characterized statistically over many pathways. For instance, the "new view" of protein folding hypothesizes that proteins fold in a multi-dimensional energy funnel by following a myriad of pathways, all leading to the same native structure. So we need efficient algorithms that can quickly explore a large number of pathways.
- A typical molecular energy function may contain many local minima, and classic simulation techniques waste considerable computation time trying to escape from these minima. They easily get

trapped in local minima, repeatedly sampling many similar conformations without obtaining much new information. Their high computational cost prevents them from being used to analyze many pathways.

In this paper, we present *Stochastic Roadmap Simulation* (SRS) as a novel computational framework to overcome both of these drawbacks [ABG+02, AGV+02]. In SRS, we build a network, called *stochastic conformational roadmap*, or just *roadmap* for short (see Figure 1 for an illustration). Such a roadmap is a directed graph, whose nodes are randomly sampled molecular conformations. Each edge between two nodes v_i and v_j in the roadmap carries a weight P_{ij}, which estimates the probability for the molecule to transition from v_i to v_j. A path between any two nodes in the roadmap corresponds to a potential motion pathway of the molecule. A roadmap thus compactly encodes a huge number of pathways. The edge probabilities determine the likelihood that the molecule follow these pathways. SRS does not trace any specific pathway on the roadmap, and thus circumvents the local minima problem encountered with the classic simulation techniques.

The probabilities attached to the edges of a roadmap directly express the stochastic nature of molecular motion. We view the motion of the molecule on the roadmap as a random walk similar to a Monte Carlo (MC) simulation run. More precisely, at each step of the random walk, a molecule either stays at the current node or moves to a neighboring node according to the assigned transition probabilities. However, to compute ensemble properties of molecular motion efficiently, we avoid performing explicit simulation runs. Instead, we treat the roadmap as a Markov chain and apply methods from the Markov-chain theory, namely first-step analysis [TK94], to process all pathways in the roadmap simultaneously, rather than one at a time as classic methods like MC simulation would do. Conceptually, this is equivalent to performing infinitely many simulation runs simultaneously and extracting statistics from them, but it results in tremendous gain in computational efficiency.

Due to the computational complexity of MC simulation, one can obtain a limited number of such simulation runs to study. However, by focusing on one pathway at a time, a MC simulation run can produce a higher density of samples along this particular one-dimensional pathway. In contrast, SRS is by necessity a coarser-grained method. It must spread the samples (the nodes of the roadmap) over the entire high-dimensional conformation space or a subset of interest. On the other hand, SRS examines many pathways at once and obtains interesting information not easily accessible by classic methods. Tests of SRS on several protein folding and ligand-protein binding examples indicate empirically that SRS computes ensemble properties satisfactorily, even with rather coarse roadmaps. In fact, some of our tests suggest that certain molecular properties can be more accurately computed by considering many coarse-grained pathways simultaneously rather than relatively few finely sampled pathways. In addition, we show formally that, with appropriately defined edge probabilities, SRS and MC simulation converge to the same sampling distribution – the Boltzmann distribution.

SRS is inspired by probabilistic roadmap (PRM) methods developed for robot motion planning [KŠLO96]. The main idea of these methods is to capture the connectivity of a geometrically complex high-dimensional space by constructing a graph of local paths connecting points randomly sampled from that space. Singh, *et al.* first introduced PRM methods to the study of molecular motion, more specifically ligand-protein binding [SLB99]. PRM methods have since been applied to protein folding as well [ADS02, ASBL01, SA01]. These earlier works treat a roadmap as a deterministic graph with heuristic edge weights based on the energy difference between molecule conformations. The heuristic weight attached to an edge measures the "energetic difficulty" of transitioning along this edge. Classic search techniques are then used to extract individual "energetically favorable" paths from the roadmap. Though similar in appearance, SRS is fundamentally different. By encoding the stochastic nature of molecular motion, our roadmap definition enables us to exploit existing tools from Markov-chain theory to analyze globally all the pathways contained in a roadmap, without distinguishing any particular ones. It also allows us to establish a formal relationship between SRS and MC simulation.

The rest of the paper is organized as follows. We first cover preliminary information regarding molecular motion simulation and Markov chains (Section 2). We then describe how to construct a roadmap (Section 3) and query it to compute ensemble properties (Section 4). We tested SRS on two types of problems. One is the computation of the probability of folding $\mathrm{P}_{\mathrm{fold}}$ (also called the *transmission coefficient*) in protein folding. At any conformation q of the protein, this parameter measures the "kinetic distance" between q and the native fold [DPG^{+}98]. The other problem is to measure the average "escape time" of a ligand from a ligand-protein binding site. The experimental results are reported in Sections 5 and 6. Section 7 discusses future work.

2 Preliminaries

2.1 Molecular modeling

The conformation of a molecule determines its 3-D structure. Conformations can be specified in various ways. For example, for a protein molecule, one may specify the positions of constituent atoms in a lattice [KS96]. In an off-lattice model, the backbone torsional angles ϕ and ψ are often used [SA01]. A simpler representation is to associate vectors to secondary structural elements and treat the angles between these vectors as the conformation parameters [ASBL01]. For ligand-protein binding, one often assumes that the protein is rigid and model the ligand with a root atom and a torsional angle for each non-terminal atom [SLB99, BSA01]. Representing the protein as non-rigid can be done, for example by identifying its main degrees of freedom [TPK02] and including them as additional dimensions of the conformation space of the ligand-protein complex. SRS is applicable to many different representations, provided that the conformation of the molecule (or the collection of molecules) is specified by a finite number of parameters that uniquely determine the 3-D position of every atom in the molecule(s). Formally, a conformation q of

d parameters is specified by a vector $(q_1, q_2, \ldots, q_d)$. The set of all conformations form the *conformation space* $\mathcal{C}$.

By determining the molecule's 3-D structure, the conformational parameters also determine the interactions between the atoms of the molecule and between the molecule and the medium, *e.g.*, van der Waals and electrostatic interactions. These interactions give rise to the attractive and repulsive forces that govern the motion of the molecule. SRS assumes that these interactions are described by an energy function $E(q)$ that depends only on the conformation q of the molecule; it does not require E to have any particular properties or functional forms.

2.2 Monte Carlo simulation

MC simulation – more precisely, the Metropolis algorithm [MRR+53] – is one of the most common techniques for studying thermodynamic properties of molecular systems. It samples the conformation space $\mathcal{C}$ of a system of molecules in order to compute quantities such as average energy and heat capacity, or the distribution of molecules in a system. A key property of MC simulation is that, in the limit, the conformation space is sampled according to the Boltzmann distribution [Lea96].

MC simulation starts at some initial conformation and performs a random walk in $\mathcal{C}$. Let q be the conformation at the current step of this random walk. To obtain the next conformation, a conformation q' is sampled from a small neighborhood of q, using a uniform or Gaussian distribution centered at q. The move to q' is accepted with a probability A that depends on the energy difference $\Delta E = E(q') - E(q)$. Define the *Boltzmann factors* $\varepsilon = \exp(-E(q)/k_{\mathrm{B}}T)$ and $\varepsilon' = \exp(-E(q')/k_{\mathrm{B}}T)$, where k_{B} is the Boltzmann constant and T is the temperature of the system. The Metropolis criterion prescribes the acceptance probability as

$$A = \begin{cases} \exp(-\Delta E/k_{\mathrm{B}}T) & \text{if } \varepsilon'/\varepsilon < 1; \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Since $\varepsilon'/\varepsilon = \exp(-\Delta E/k_{\mathrm{B}}T)$, the condition $\varepsilon'/\varepsilon < 1$ holds if and only if $\Delta E > 0$. So, if a move decreases the energy, it is always accepted; otherwise, it is accepted with probability $\exp(-\Delta E/k_{\mathrm{B}}T)$. If the move from q to q' is accepted, the simulation transitions to q'; otherwise, it stays at q. The procedure repeats to generate a series of sampled conformations, until some termination condition is satisfied (*e.g.*, the maximal number of steps has been achieved, or the quantity being computed stabilizes).

This simulation procedure guarantees that if the number of simulation steps becomes sufficiently large, the sampled conformations are distributed according to the Boltzmann distribution:

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

where $Z_\beta = \int_{\mathcal{C}} \exp(-E(q)/k_{\mathrm{B}}T)\, dq$ is the normalization constant. So any subset $S \subseteq \mathcal{C}$ is sampled with probability

$$\beta(S) = \int_S \beta(q)\, dq.$$

MC simulation is also an important tool to study molecular motion [SKS01, KS96]. However, it is computationally intensive. Each simulation run yields a series of sampled conformations defining a single pathway. Due to the high potential variance between independent runs, the simulation must be run many times over extended durations in order to produce accurate statistical results. Moreover the energy function E typically contains many local minima. A simulation run spends most of its time overcoming energy barriers to escape from these local minima. Many similar conformations are sampled near the same local minimum, without generating new information.

2.3 Stationary distribution of a Markov chain

A Markov chain is a stochastic process that takes values from a finite or countable set of states $s_1, s_2, \ldots, s_n$. The probability P_{ij} of going from state s_i to s_j depends only on states s_i and s_j. Under suitable conditions, a Markov chain has an associated limit distribution $\pi = (\pi_1, \pi_2, \ldots)$ that can be obtained as follows. Starting at an arbitrary initial state, perform a random walk over the set of states. At each step of this walk, make a move to the next state with the transition probability P_{ij}. If we let the walk continue infinitely, then under the condition that the Markov chain is *ergodic*, each node v_i is visited with a fixed probability π_i in the limit, regardless of the starting node [TK94]. So π describes the limit behavior of *all* possible random walks. The probability π_i gives the fraction of the time that v_i is visited in the limit.

The limit distribution π satisfies 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 that $\pi_i \geq 0$ for all i and $\sum_i \pi_i = 1$, the solution to Eq. (2) is guaranteed to be a well-defined probability distribution. Eq. (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*.

If the conformation space of a molecule is discretized into a finite set of states, MC simulation over this space can be described by a Markov chain with appropriately defined transition probabilities. The stationary distribution of the Markov chain then gives the limit behavior of the MC simulation.

3 Stochastic conformational roadmaps

In SRS, we preprocess molecular pathways by precomputing a roadmap that provides a discrete representation of molecular motion. A roadmap compactly encodes a large number of MC simulation paths simultaneously and enables us to perform key computation efficiently.

3.1 Roadmap construction

A roadmap G is a directed graph. Each node of G is a randomly sampled conformation in $\mathcal{C}$. Each (directed) edge from node v_i to node v_j carries a weight P_{ij}, which represents the probability that the molecule will move to conformation v_j, given that it is currently at v_i. The probability P_{ij} is 0 if there is no edge from v_i to v_j. Otherwise, it depends on the energy difference $\Delta E_{ij} = E(v_j) - E(v_i)$.

To construct a roadmap, our algorithm first samples conformations independently at random from $\mathcal{C}$. In our current implementation, we use the uniform distribution by picking a value for each conformational parameter $q_1, q_2, \ldots$ uniformly at random from its allowable range (see Section 7 for a discussion of more efficient sampling strategies). Next, for each node v_i, the algorithm finds its nearest neighbors using a suitable metric such as the RMS distance [Lea96]. It then creates an edge between v_i and every neighboring node v_j and attaches to it the transition probability P_{ij} defined by

$$P_{ij} = \begin{cases} \frac{1}{d_j} \exp(-\Delta E_{ij}/k_{\mathrm{B}}T) & \text{if } \frac{\varepsilon_j/d_j}{\varepsilon_i/d_i} < 1; \\ \frac{1}{d_i} & \text{otherwise;} \end{cases} \tag{3}$$

where ε_i and ε_j are the Boltzmann factors at v_i and v_j, and d_i and d_j are the number of neighbors of v_i and v_j. If there is no edge between v_i and v_j, then they are considered too far apart for their energy difference to be a good basis for estimating the transition probability, and we set $P_{ij} = 0$. The molecule can still move from v_i to v_j, but the move necessarily traverses one or several other nodes of the roadmap. Finally, a self-transition probability $P_{ii} = 1 - \sum_{j \neq i} P_{ij}$ is attached each node v_i, thus ensuring that the transition probabilities from any node sum up to 1. We retain the roadmap only if it contains a single connected component.

In contrast to the heuristic edge weights used in [SLB99, ASBL01, SA01], these transition probabilities enable us to establish a formal relationship between SRS and MC simulation [ABG+02]. We now describe this important relationship.

3.2 Relationship with Monte Carlo simulation

A MC simulation run is a random walk in the conformation space $\mathcal{C}$. We can perform a similar walk on the roadmap G as follows: at node v_i of G, we choose a node v_j uniformly at random from the set of neighbors of v_i and propose a move to v_j. The move is accepted with probability

$$A_{ij} = \begin{cases} \frac{d_i}{d_j} \exp(-\Delta E_{ij}/k_{\mathrm{B}}T) & \text{if } \frac{\varepsilon_j/d_j}{\varepsilon_i/d_i} < 1; \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

Expressions (1) and (4) are similar, except for the additional factor d_i/d_j. This factor is needed because, while the neighborhoods of all sampled conformations in MC simulation have the same size, the number of neighbors varies from one node to another for a random walk on the roadmap. Since node v_i has d_i neighbors and each one is chosen with probability $1/d_i$, the transition probability from v_i to v_j is $(1/d_i)A_{ij}$, which, after simplification, is equal to P_{ij} given in (3). Hence, with our choice of transition probabilities, every path in the roadmap corresponds to a MC simulation run.

We have also mentioned in Section 2.2 that MC simulation generates sample conformations with a distribution that converges to the Boltzmann distribution β. So, in the limit, the probability of sampling any subset $S \subseteq \mathcal{C}$ is

$$\beta(S) = \frac{1}{Z_\beta} \int_S \exp(-E(q)/k_{\mathrm{B}}T)\, dq.$$

Now we would like to ask the same question for SRS. What is the limit behavior of SRS? In other words, if we perform an arbitrary long random walk on the roadmap as described above, what is the probability of sampling a subset $S \subseteq \mathcal{C}$? Since, by construction, a roadmap is connected, it defines an ergodic Markov chain with transition probabilities P_{ij} [TK94]. So, the limit behavior of SRS is governed by the stationary distribution of this Markov chain, given by the following lemma:

Lemma 1 *A stochastic conformational roadmap defines a Markov chain with stationary distribution*

$$\pi_i = \frac{1}{Z_\pi} \exp(-E(v_i)/k_{\mathrm{B}}T) \quad \textit{for all } i, \tag{5}$$

where $Z_\pi = \sum_i \exp(-E(v_i)/k_{\mathrm{B}}T)$ *is a normalization constant.*

Proof: See Appendix A. □

To estimate the probability of sampling a set S, we simply sum the stationary distribution π over all the nodes v_i that lie in S:

$$\pi(S) = \sum_{v_i \in S} \pi_i = \frac{1}{Z_\pi} \sum_{v_i \in S} \exp(-E(v_i)/k_{\mathrm{B}}T).$$

If SRS represents the stochastic motion of a molecule with the same limit behavior as MC simulation, then we expect the limit distributions of these two methods to converge. In other words, $\pi(S)$ should approximate

$\beta(S)$ to any arbitrary precision, given a suitably dense roadmap. This is formally summarized in Theorem 1. In the appendix, we provide a complete statement of the theorem.

Theorem 1 *Let S be any subset of the conformation space $\mathcal{C}$ with relative volume $\mu(S) > 0$. For any $\varepsilon > 0$, $\delta > 0$, and $\gamma > 0$, a roadmap with N uniformly sampled nodes (where N is polynomial in $\ln(1/\gamma)$, $\| \exp(-E(v)/k_{\mathrm{B}}T)\|_S$, $1/\mu(S)$, the normalization constant Z_β, $1/\varepsilon$ and $1/\delta$), the difference between the probability $\beta(S)$ and the estimate $\pi(S)$ from the roadmap is bounded by:*

$$(1-\delta)\beta(S) - \varepsilon \leq \pi(S) \leq (1+\delta)\beta(S) + \varepsilon, \tag{6}$$

with probability at least $1-\gamma$, where $\|f\|_S = \sup_v f(v) - \inf_v f(v)$ and $Z_\beta = \int_{\mathcal{C}} \exp(-E(q)/k_{\mathrm{B}}T)\, dq$.

Proof: See Appendix B. □

The 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. In particular, for any subset S of $\mathcal{C}$, the theorem tells us that there exists N such that if we sample N uniformly distributed nodes in the roadmap, and find the points falling into S, the sum of the stationary distribution on this subset of points will converge to the Boltzmann distribution β in S. Since MC simulation also approaches β in the limit, it follows that both SRS and MC simulation converge to the same limit distribution.

Figure 2 illustrates the result of Theorem 1. It shows that the error in our roadmap estimates of the stationary distribution decreases as the size of the roadmap increases, as predicted by the theorem. The plot was obtained by evaluating our roadmap estimates of stationary distribution on a fictitious energy landscape in a 2-D conformation space. We divided the space 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 MC integration. Figure 2 shows the average error in our estimates, *i.e.*, $(1/100)\sum_{i=1}^{100} |\pi(B_i) - \beta(B_i)|$.

Furthermore, Theorem 1 deals with the asymptotic 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 factor $\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S$, the normalization constant Z_β, and the inverse $1/\mu(S)$ of the relative volume of S, where $\mu(S)$ is defined as the ratio of the volume of S to the volume of $\mathcal{C}$. Although this bound demonstrates the polynomial convergence of our algorithm, in practice this bound may be overly pessimistic and our convergence may be faster, as suggested by the applications presented in this paper.

The above result establishes an important link between SRS and MC simulation. The limiting distribution of SRS on any subset of the conformational space is Boltzmann. Furthermore, every ensemble property that can be computed by SRS can also be computed by averaging from many MC simulation runs, assuming unlimited computation time. The novelty of the SRS framework is the way computation is organized. The precomputation of a roadmap and the subsequent exploitation – or *query* – of this roadmap result in major computational gains, as demonstrated in the rest of this paper.

4 Roadmap query

A roadmap G encodes considerable information on molecular motion. For instance, given two nodes v_i and v_j in G, we could 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 would lead to results similar to those presented in [SLB99, ASBL01, SA01]. However, since a roadmap explicitly captures the stochastic nature of molecular motion, it allows us to take advantage of powerful tools from the Markov-chain theory. We now focus on one such tool, known as *first-step analysis*.

To illustrate our description, consider a roadmap G built in the conformation space of a protein. Assume that the native fold of this protein is known. Let $\mathcal{F}$ stand for the set of nodes in G that are structurally similar to the native fold. For instance, $\mathcal{F}$ may consist of all the nodes that lie within some given distance from the native fold. It is an example of a *macrostate*, an abstraction that combines a set of nodes into a single entity; we refer to it as the folded state. Assume we are interested in knowing, for every node v_i in G, the expected number of transitions, t_i, that it takes to go from v_i to the folded state, *i.e.*, any node in $\mathcal{F}$. A naive method to compute t_i would be to perform many MC simulation runs, starting from v_i, and average the number of transitions taken by each run. As mentioned before, this method would require performing many simulation runs for each node v_i in order to get a reasonably accurate value of t_i.

Instead, first-step analysis proceeds by conditioning on the first transition. 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 either enter the folded state or reach 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 t_j. So, we get the following system of linear equations:

$$t_i = 1 + \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 0 + \sum_{v_j \notin \mathcal{F}} P_{ij} \cdot t_j \quad \text{for every } v_i \notin \mathcal{F}. \tag{7}$$

In the second term of (7), P_{ij} is multiplied by zero, because we stop as soon as we enter the folded state. See Figure 3 for an illustration.

The linear system (7) contains one equation and one unknown for each node $v_i \notin \mathcal{F}$. By solving this system, we obtain t_i for all the nodes simultaneously, without performing any explicit simulation.

To solve the linear system (7), we rewrite it in matrix form:

$$(\mathbf{I} - \mathbf{Q}) \cdot \mathbf{t} = \mathbf{b}, \tag{8}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{Q}$ is a matrix whose entries are the transition probabilities P_{ij}, $\mathbf{t}$ is the vector of unknowns $t_i, i = 1, 2, \ldots$, and $\mathbf{b}$ is a vector collecting the remaining constant terms in (7). Since a roadmap usually contains many nodes, the size of $\mathbf{I} - \mathbf{Q}$ is large, so direct methods for solving (8), *e.g.*, Gaussian elimination, are impractical. However, the ergodicity of the Markov chain defined by the roadmap guarantees that a unique solution to (8) exists. So iterative methods can instead be used. In particular, the naive iteration

$$\mathbf{t}^{(k+1)} = \mathbf{Q} \cdot \mathbf{t}^{(k)} + \mathbf{b}$$

converges to the unique solution. This iterative method amounts to performing many simulation runs simultaneously using matrix multiplication. More efficient iterative methods, such as the conjugate-gradient method [Saa96], can also be used. Furthermore, since every node in the roadmap is directly connected to a relatively small number of neighboring nodes, $\mathbf{Q}$ is a sparse matrix. Sparse-matrix ordering algorithms greatly reduce the running time of iterative solvers [GL89, GMS92].

5 Computing the probability of folding

In this and next section, we describe the application of SRS to compute two specific ensemble properties: the *probability of folding* in protein folding and the *escape time* in ligand-protein binding.

Protein folding is one of the most marvelous processes in nature. Under suitable conditions, most proteins go through a series of geometric transformations and arrive at the native folds where they perform intricate biological functions. There are large on-going efforts to understand the folding process (*e.g.*, [Tea01, SP00]): What geometric transformations does a protein go through during folding? Which conformations are "closer" to the native fold along the folding pathways?

To address this type of questions, the probability of folding ($\mathrm{P_{fold}}$) – also known as the transmission coefficient – has been introduced to measure how far away a protein conformation is from the native conformation kinetically [DPG+98]. For a folding process dominated by two stable states, a folded state $\mathcal{F}$ and an unfolded state $\mathcal{U}$, the $\mathrm{P_{fold}}$ value τ for a conformation q is the probability of reaching $\mathcal{F}$ before $\mathcal{U}$, starting from q. If $\tau > 0.5$, then the protein is more likely to fold first than to unfold first, and therefore q is kinetically closer to the folded state. Trivially, if q is in $\mathcal{F}$, then $\tau = 1$, and if q is in $\mathcal{U}$, then $\tau = 0$. The $\mathrm{P_{fold}}$

value at q is not associated with any particular folding pathway, but depends on all possible pathways from q. It thus describes the average behavior of the folding process. In this sense, it is an ensemble property.

5.1 First-step analysis

Using SRS, we can compute P_{fold} as follows. Let $v_i, i = 1, 2, \ldots$ be the nodes of the computed roadmap, and τ_i be the P_{fold} value for v_i. First-step analysis yields the following equation for every node v_i not in $\mathcal{F}$ or $\mathcal{U}$:

$$\tau_i = \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 1 + \sum_{v_j \in \mathcal{U}} P_{ij} \cdot 0 + \sum_{v_j \notin (\mathcal{F} \cup \mathcal{U})} P_{ij} \cdot \tau_j. \tag{9}$$

Eq. (9) is obtained by conditioning on the first transition. After one step of transition, we have three possibilities:

1. We reach a node in $\mathcal{F}$. Then, we have reached $\mathcal{F}$ before $\mathcal{U}$ with probability 1.
2. We reach a node in $\mathcal{U}$. Then, we have reached $\mathcal{U}$ before $\mathcal{F}$, and the probability of reaching $\mathcal{F}$ before $\mathcal{U}$ is 0.
3. We reach a node v_j not in $\mathcal{F}$, nor in $\mathcal{U}$. The value τ_i then depends on the value of τ_j.

Linear system (9) has the same matrix form as the example in Section 4. A unique solution exists and can be obtained by an iterative solver.

We can improve the accuracy and potentially the speed of the iterative solver by setting all the self-transition probabilities in the roadmap to 0 and renormalizing the other probabilities. Set

$$\begin{aligned} P'_{ii} &= 0 && \text{for all } i, \\ P'_{ij} &= P_{ij} / \textstyle\sum_{k \neq i} P_{ik} && \text{for all } i \neq j \end{aligned} \tag{10}$$

and solve the linear system

$$\tau_i = \sum_{v_j \in \mathcal{F}} P'_{ij} \cdot 1 + \sum_{v_j \in \mathcal{U}} P'_{ij} \cdot 0 + \sum_{v_j \notin (\mathcal{F} \cup \mathcal{U})} P'_{ij} \cdot \tau_j. \tag{11}$$

If we think in terms of performing a random walk on the roadmap as described in Section 3.2, then setting the self-transition probabilities to 0 is equivalent to accepting all proposed moves. It is easy to verify that the linear systems (11) and (9) have the same solution by substituting (10) into (11). However, if we write (11) in the matrix form, the matrix $\mathbf{I} - \mathbf{Q}$ contains 1 in all its diagonal entries, which are greater than or equal to the corresponding entries in the matrix for (9). So (11) tends to be a better conditioned system for iterative methods.

5.2 Experimental results

We now show results on three examples. The first example is based on a relatively simple energy function in a fictitious 2-D conformation space. The other two examples study real proteins. We compare the results obtained with SRS to those obtained with MC simulation, and demonstrate that SRS reduces the running time by several orders of magnitude, while being more accurate. The main reason for using fictitious data in the first example is that MC simulation takes extremely large amounts of computation time on real proteins. The simpler energy function in this example makes it possible to perform more extensive comparison than is practically possible in the other two examples.

In our current implementation, the roadmap construction part of SRS is coded in C++ and the linear system solver in Matlab. MC simulation is implemented entirely in C++. Timing results reported below were obtained on a 1GHz Pentium-III PC with 1GB of memory.

5.2.1 Example in 2-D space

In this example, the "energy" function E is constructed as a linear combination of radially symmetric Gaussians over a 2-D space, with a paraboloid centered at the origin (Figure 4). The centers, the decay rates, and the heights of the Gaussians are picked at random. The energy varies between roughly -5 and 5. There are two local minima, with energies -4.88 and -4.98, corresponding to the folded and unfolded states. They are at (0,0) and (-50,-50) respectively. The landscape goes from -100 to +100 along both axes. The folded and unfolded macrostates are regions of radius 1 around the minima. The maximum step size in MC simulation is ± 2, i.e., 0.01 in each dimension if the space is normalized to be between 0 and 1. In this 2-D space, we use the Euclidean metric for finding neighboring nodes of the roadmap.

We first used SRS to compute $\mathrm{P_{fold}}$ for 100 sampled conformations with a roadmap of approximately 10,000 nodes. We then used MC simulation to compute $\mathrm{P_{fold}}$ for the same conformations. In MC simulation, we performed 500 independent runs for each node. Each run stops as soon as it enters 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 in Figure 5. All the points in the plot lie close to the diagonal line, indicating that the results from the two methods are in good correspondence.

We conducted further tests by varying the number of nodes sampled by SRS and the number of independent MC simulation runs per node. In each test, we summarize the correspondence between the results from the two methods by their normalized correlation coefficient, which is defined as

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

for two vectors x and y, where $\langle \cdot \rangle$ denotes the operation of taking the average. Note that the magnitude of κ is always between 0 and 1, with 0 indicating no correlation and 1 indicating perfect correlation. Figure 6 shows the results of these additional tests. The number of nodes in the roadmap is indicated along the horizontal axis and the correlation coefficient κ along its vertical axis. The graph contains three curves, respectively corresponding to 100, 500, and 1000 independent MC simulation runs per node. The three curves show a generally similar trend. Initially κ 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 indicates perfect correlation. Since κ measures only the correspondence 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 MC simulation. However, we can get a hint by comparing the three curves. For a roadmap of a given size, κ generally improves as we increase the number of independent MC runs per node. This seems to indicate that SRS gives the more accurate results: when the number of independent MC runs per node increases, the variance of MC simulation decreases, and simultaneously, the results get closer to those obtained from SRS.

In a typical run, SRS took about 8 seconds to construct a roadmap of 10,000 nodes, and 3 seconds to solve the linear system and obtain $\mathrm{P_{fold}}$ values for all the nodes. The running time of MC simulation is proportional to the product of the number of conformations for which $\mathrm{P_{fold}}$ is computed by the number of runs per conformation. The time needed to perform 100 runs at each of the 10,000 nodes of the roadmap, hence 10^6 runs, is around 800,000 seconds. However, this comparison of running times is of limited interest, since the cost of computing our fictitious energy function is much smaller than that of computing any reasonable energy function for a real protein.

5.2.2 1ROP and 1HDD

More interestingly, we also tested SRS on two real proteins (Figure 7), repressor of primer and engrailed homeo-domain, which are identified as 1ROP and 1HDD, respectively, in the Protein Data Bank [B$^+$77]. 1ROP is a dimer made of two identical monomers, each containing 56 residues forming two α helices connected by a loop. As in [STD95], we study a single monomer in isolation. Protein 1HDD contains 57 residues forming three α helices packed against each other.

Our implementation encodes the conformation of a protein with the vector-based model previously used in [SB97, ASBL01]. In this representation, a protein is described as a sequence of vectors, each associated with a secondary structure element. In our model, 1ROP has 6 degrees of freedom (DoF), and 1HDD has 12 DoF. Our energy function uses the H-P model [STD95], which consists of two terms measuring the hydrophobic interaction and the excluded volume, respectively. As in [ASBL01], amino acids are categorized into two groups, hydrophobic (H) and hydrophilic (or polar, P). H-H contacts are favorable, whereas H-P or

P-P contacts do not contribute to the energy. The exclusion term ensures that no two atoms are too close. These terms are a function of the distances between side-chain centroids, for the conformation of interest. This model assumes that hydrophobic interactions drive the folding process and that the specific identity of the side-chains is only responsible for the fine-tuning of the fold.

To find the nearest neighbors of a node, we use the cRMS metric, which is defined as follows: For two conformations of a protein, P and Q, given their C-α or sidechain centroid coordinates p_i and q_i, $i = 1, \ldots, n$, $cRMS(P, Q) = min_T \sqrt{\frac{1}{n} \sum_i \|p_i - Tq_i\|^2}$ where T is a matrix denoting rigid body transformation(rotation and translation). Given P and Q, there is a closed form solution for T. We use Bioinformatics Template Library[1] to compute T.

In both SRS and MC simulation, we discard conformations that cause steric hindrance, *i.e.*, self-collision of atoms in the protein. We define the folded state to contain all conformations within a small cRMS distance of the native fold (3 Å for 1ROP and 5 Å for 1HDD), and the unfolded state to contain all the conformations within 10 Å of the fully extended conformation. The roadmap construction algorithm uses the RMS distance to find neighboring nodes, as it better measures the similarity between two protein conformations than the Euclidean distance in conformation space. The maximum step size in MC simulation was ± 0.05 in the normalized coordinates, for each DoF. The maximum cRMS distance to connect two nodes was 5 Å and 7 Å for 1ROP and 1HDD, respectively.

We computed the P_{fold} values at about 45 randomly selected conformations for both 1HDD and for 1ROP, using both SRS and MC simulation. With SRS, we computed the estimates with roadmaps having increasing numbers of nodes. In MC simulation, we performed up to 300 independent runs at each of the selected conformations. We then computed the correlation coefficient of the computed values as a function of the size of the roadmap, for three numbers of MC runs per node (100, 200, and 300). The results, given in Figure 8, suggest conclusions similar to those derived from the fictitious energy function in the

[1] http://people.cryst.bbk.ac.uk/ classlib/bioinf/BTL99.html

previous section. First, SRS estimates generally improve very fast as the roadmap size increases. Second, the correlation tends to increase as we perform more MC runs per node. We also tried MC simulation runs with larger step sizes and observed that as the step size increases, the correlation between the results obtained from MC simulation and SRS worsens. We compared the P_{fold} values obtained from the two methods using their average absolute differences as well, instead of their correlation coefficients; obtaining similar results.

The total time to generate a roadmap with 5,000 nodes and compute the P_{fold} values for *all* these nodes was about 1.5 hours on a 1GHz Pentium-III PC with 1GB of memory. In comparison, it took an average of five to six hours of computation time in order to execute 300 MC simulation runs required to estimate P_{fold} at just *one* conformation for 1ROP (about 10 hours for 1HDD). To compute P_{fold} at the 49 selected conformations of 1ROP, MC simulation took about 270 hours. Hence, SRS produces similar results by at least four order of magnitude faster in these examples.

6 Analysis of ligand-protein interactions

Ligand-protein binding is another important biological process, in which a small molecule, the *ligand*, attaches itself to a specific site, usually a cavity on the surface of a larger receptor protein in order to inhibit or enhance activities at the site. A protein often has several cavities where a ligand could potentially bind. We refer to them as *potential binding sites*. The computational analysis of ligand-protein binding has already attracted considerable attention (see, *e.g.*, [MGH+98, WKK99]).

Let us consider the conformation space $\mathcal{C}$ of a ligand-protein complex with a suitably defined energy function. A bound conformation $q \in \mathcal{C}$ generally corresponds to a local energy minimum and has a "funnel of attraction" around q to stabilize the ligand. Following [CV01], we define the funnel of a bound conformation q as the set of all conformations within 10 Å of q in RMSD. Figure 9 shows funnels of attraction for four potential binding sites on lactate dehydrogenase.

An interesting measure of affinity of a ligand to a potential binding site is the expected amount of time the ligand would take to escape the funnel of attraction of this site. At the catalytic (or active) site, the ligand is usually bound with very high affinity. So, it should take much longer to escape from this site's funnel, than from the funnels of other potential binding sites. Similarly, lowering the affinity of a protein to the ligand (for instance, by mutating a residue in the catalytic site) should result in a faster escape of the ligand from the catalytic site. With MC simulation, a natural choice to estimate the ligand's escape time is to count the number of simulation steps:

Definition 1 *The* escape time τ *from a potential binding site* v *is the* expected *number of MC simulation steps, starting from* v*, required for the ligand to reach a conformation outside the funnel of attraction* $\mathcal{A}$ *of* v*.* □

Below we use SRS to estimate the escape time defined as above.

6.1 Ligand-protein modeling

We represent the ligand-protein complexes as in [SLB99, ASBL01, AGV$^+$02]. The protein is considered rigid, while the ligand is flexible. One atom in the ligand is designated to be the base and is assigned 5 DoF relative to a coordinate system attached to the protein; an additional torsional DoF is associated with each other non-terminal atom. Rings are assumed rigid and are assigned no DoF. Bond angles and lengths are considered constant. The ligand's set of DoF define the parameters of a conformation of the ligand-protein complex.

To calculate the energy of interaction between the ligand and the protein, as well as the internal energy of the ligand, we used a potential function that incorporates electrostatic and van der Waals components. Since the standard Coulombic equation of electrostatic interaction is valid only for an infinite medium of uniform dielectric, it cannot be used here. The dielectric discontinuity between protein and solvent generates induced or reflected charges that can play a significant role in the binding process. Hence, we modeled electrostatics using the Poisson- Boltzmann equation, which is a widely accepted model of electrostatic interactions in solution and models solvent and ionic effects.

We used the Delphi program [SH90] to solve the equation on a 3-D grid at a resolution of either 1A or 0.5A. The van der Waals potentials are computed at the same grid resolution by calculating for each grid point the potential contribution of all receptor atoms within a threshold distance of 10Å.

We compute the energy of interaction of every ligand atom with the protein by indexing the atoms center to the nearest grid point and retrieving the van der Waals and electrostatic potentials at this point. The total energy of interaction is computed by summing the contributions of each atom. The ligands internal energy is computed by applying the standard van der Waals and Coulombic equations to each non-bonded pair of ligand atoms. Since a ligand is small and flexible, we assume that its surface is not well defined and hence use the standard Coulombic equation, with a dielectric constant between 60-80. The charges on each atom of each ligand were computed as a formal charge taking into account resonance structures of the molecule.

6.2 First-step analysis

We first construct a roadmap G over the ligand-protein conformation space. We then apply first-step analysis to obtain a system of equations almost identical to (7). Let $\mathcal{A}$ be the set of nodes in G that lie in the funnel of the bound conformation q_{b}. Let t_i be the expected number of transitions to reach a conformation outside of $\mathcal{A}$, starting from a node $v_i \in \mathcal{A}$. We have

$$t_i = 1 + \sum_{v_j \notin \mathcal{A}} P_{ij} \cdot 0 + \sum_{v_j \in \mathcal{A}} P_{ij} \cdot t_j \quad \text{for every } v_i \in \mathcal{A}. \tag{12}$$

The solution of the above equations gives an estimate of the escape time for every node in the funnel, including the bound conformation q_{b}. We define the average escape time from the funnel is the escape time from q_{b}.

6.3 Analyzing the effects of mutations

We first applied SRS to analyze the effects of mutations in the catalytic site of a protein on the escape time of a ligand.

6.3.1 Computational mutagenesis

Computational mutagenesis is a new and exploratory area of computer-aided protein design. It is based on the biological method of site-directed mutagenesis. A few amino acids are either deleted entirely or replaced by other amino acids, or alternatively, the side chains of amino acids are altered. Site-directed mutagenesis has proven quite useful for many studies, including substrate recognition and identification of catalytic amino acids [CWC+86]. The mutations made through this method are specific in terms of what changes are made, local in terms of exactly which amino acids are affected, and sound in terms of having no significant structural ramifications. Computational mutagenesis embodies these concepts from site-directed mutagenesis, but enables mutations to be performed *in silico* providing the obvious benefits of speed and ease at perhaps the expense of model accuracy. Reyes and Kollman, for example, have shown encouraging early results in utilizing computational mutagenesis to study binding specificity [RK00].

6.3.2 Mutagenesis study on lactate dehydrogenase

Here, we employ computational mutagenesis in order to study the sensitivity of SRS when applied to the analysis of ligand-protein interactions by computing escape times from funnels. In one series of tests, we used oxamate (an inactive analogue of pyruvate) and lactate dehydrogenase.

Lactate dehydrogenase (LDH) LDH is a well-studied enzyme [CWHH85, Har89] that, when bound to its coenzyme NADH, is able to catalyze the reduction of pyruvate to lactate. LDH has been proposed as a general framework on which to design and synthesize new enzymes [DWH+91]. We use dogfish apolactate dehydrogenase (PDB: 1LDM) and oxamate (an analog of pyruvate) as a model on which to perform computational mutagenesis.

The active site of LDH is well understood. The chemical environment of oxamate in its bound conformation in the LDH-NADH-substrate complex is depicted in Figure 10. The amino-acids that play a significant role in the catalytic activity of the enzyme are shown. Arg169 assists in orienting and binding the substrate [HCW+87]. Arg106 polarizes the carbonyl bond on the substrate [CWC+86]. His193 is an important catalytic residue, which donates a proton to the substrate during its reduction [HLSR75]. His193 is then stabilized by Asp166 [CBA+88]. In native LDH, before the binding of the coenzyme or the substrate, a loop of polypeptide chain (residues 97 to 107) is positioned away from the active site. After the binding of coenzyme and the substrate, a rearrangement in protein structure is induced which results in the loop being positioned over the active site as shown in Figure 10.

Mutations Two sets of mutations were performed on LDH based largely on prior *in vitro* work [DWH+91]. The first set consisted of changing charged and catalytic amino acids (His193 → Ala, Arg106 → Ala, and both His193 → Ala and Arg106 → Ala). These mutants cause a large reduction in the energetic structure of the active site, thus, can provide insights into the sensitivity of SRS to coarse changes in the system. The second set of mutants (Asp195 → Asn, Gln101 → Arg, Thr245 → Gly) play a cursory role in catalysis and thus were expected to have a less significant effect. This second set of mutants, on the other hand, can provide us with insights into the sensitivity of SRS to fine changes in the system, as they cause small or no reduction in the energetic structure of the active site.

Mutations were performed using Sybyl (distributed by Tripos Inc.). No structural re-calculation or minimization was performed, hence assuming as in [RK00] that the structural change upon mutation is insignificant. We computed 20 roadmaps for every mutation. The roadmaps generated contained 10,000 nodes uniformly sampled in a region within 15 Å in RMSD of the bound conformation.

Our results are summarized in Table 1. The variations of the average computed escape times relative to wild type (given in column 3) agree with biologically expected changes, as discussed below and summarized in column 4 of the table.

His193 → Ala His193 is an important catalytic and charged amino acid. Replacing His193 with Ala would cause a significant reduction in the energetic structure of the active site [WHF+88], which results in less tight binding between enzyme and substrate. Therefore, decreasing the affinity of the substrate for the enzyme. We would expect a faster escape from the bound conformation.

Arg106 → Ala Arg106 is also an important catalytic and charged amino acid. Similar to His193, we would expect a significant reduction in the energetic structure of the active site [WHF+88], which would lead to a reduced affinity between enzyme and substrate. Thus, the substrate would be able to escape in less time from the bound conformation when compared to wild type.

His193 → Ala and Arg106 → Ala Both His193 and Arg106 are necessary catalytic and charged amino acids for enzymatic function of LDH. Thus, their replacement with Alanine would result in a significant reduction in energetic structure of the chemical environment of the LDH-substrate-complex [WHF+88]. Therefore, we would expect the substrate to quickly escape from the active site.

Asp195 → Asn Asp195 likely plays a significant role in charge conservation by providing a negative charge. Thus, its replacement with the neutral Asn would likely affect the energetic structure of the active site [WHF+88] by increasing the affinity of the substrate for the active site. This would result in slower escape for the substrate.

Gln101 → Arg Gln101 plays an important role in loop movement [WHF+88]. Recall that binding of NADH and substrate induces a conformational change on the loop region causing it to close over the active site. Gln101 is replaced by Arg which is a positively charged amino acid, however, the location of the mutation is on the outside of the loop, therefore the additional charge can be assumed to be negligible when computing escape time. Furthermore, since our LDH is held rigid in these experiments, the Gln101 → Arg mutation is not expected to cause significant change in escape times.

Thr245 → Gly Thr245 employs a large side chain and thus reduces the total volume of the active site. In order to increase the volume of the active site without causing significant energetic restructuring of the active site, Thr245 was replaced by Gly, which has a much smaller side chain resulting in a net increase in total volume of the active site [WHF+88]. Thus, escaping should become easier for the substrate.

6.4 Predicting the active site

A receptor protein may have several potential binding sites. Therefore, it is important to be able to predict which is the active site, the site that enables specific biological functions, *e.g.*, inhibition or catalysis. We hypothesize that due to higher energy barriers, longer escape time results from the funnel of attraction of the active site and may serve as a basis for prediction.

We applied our method to seven different ligand-protein complexes whose active sites are known. They are listed in Table 2. For each complex, the number of DoF of the ligand is listed in column 3 of the table.

To find potential binding sites, we picked random conformations and performed energy minimization from them. In the end, in addition to the true bound conformation, we retained four obtained conformations as the potential binding conformations, based on their energies (they must be among the lowest), their distance to the protein surface (the distance between the ligand's center of gravity and the closest protein atom center should be less than 5Å), and their distance from each other (any two binding site must be further apart than 10Å RMSD).

We computed 20 roadmaps for every potential binding site. Each roadmap had 10,000 nodes. These nodes were uniformly sampled in a region within 15 Å in RMSD of the bound conformation. We then solved for the escape times using equation (12). The averaged results are listed in Table 3. Every row of the table shows the escape-time estimates for the various binding sites of a ligand-protein complex.

In four of the seven cases, the escape time for the active site is larger (escape is slower) than those for the other binding sites by at least two orders of magnitude, clearly distinguishing the active site. In two other cases (1LDM and 1CJW), the escape time for the active site is close to the largest. In one case (1AID), the escape time fails to give a clear indication on the active site. This failure may have several causes. The size of the roadmaps may be too small to estimate the escape times accurately. The energy function that we use may not be detailed enough to capture all significant interactions between the ligand and the protein. Finally, it is possible that the active site may not always have the highest escape time in nature.

For each binding site, our software took about 7 minutes (on a 1GHz Pentium-III PC with 1GB of memory) in total to construct the roadmap and solve the linear systems yielding the escape-time estimates.

7 Conclusion and future work

Stochastic Roadmap Simulation is a new computational framework for analyzing molecular motion and computing ensemble properties of such motion. It is closely related to MC simulation. Each path in a stochastic conformational roadmap can be interpreted as an MC simulation run. Furthermore, we can show that SRS converges to the same sampling distribution as MC simulation. A salient feature of SRS is that it compactly encodes many motion pathways. Unlike classic Monte Carlo and molecular dynamic methods, which study one pathway at a time, SRS processes multiple pathways simultaneously. As a result, SRS avoids the local-minima problem that plagues the existing methods and achieves tremendous gains in computational efficiency, as demonstrated in Section 5. Thus, SRS enables computational studies that would otherwise be impractical.

We tested SRS on two interesting biological problems. In the first problem, we computed the probability of folding, which measures the "kinetic distance" between a protein conformation and the native fold. Our experiments on a synthetic energy landscape and on two real proteins show that SRS reduces the running times by several orders of magnitude, while obtaining arguably more accurate results, when compared to MC simulation.

In the second problem, we computed estimates of the expected time for a ligand to escape from the funnel of attraction of a binding site. This estimate was used to measure the effects of mutations on the catalytic site of an enzyme. We observed biologically expected changes in escape time, such as a faster escape when a neutral amino acid replaced a charged one responsible for orienting the ligand. We also used escape time to distinguish the active site of a protein from other potential binding sites on several real ligand-protein complexes.

In [DPG+98], Du *et al.* suggest that the probability of folding (also called 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 probability of folding viable, which would potentially enable its use in practice.

Nevertheless, several questions still need to be explored. The most important and interesting algorithmic question is to develop sampling strategies that will make it possible to study larger molecules with more complex energy models. Currently, we sample the conformation space $\mathcal{C}$ or a selected subset of it uniformly at random. As the dimension of $\mathcal{C}$ increases, it becomes more difficult to obtain biologically interesting conformations with uniform sampling, and the quality of results obtained from uniformly sampled roadmaps is likely to degrade. In contrast, provided there are few well defined pathways, MC simulation would follow them to reach the active site or the folded state quickly even though the dimension of $\mathcal{C}$ may be high.

One approach to address the dimensionality problem in SRS is to construct a sampling distribution that favors low-energy conformations over high-energy ones, as molecules are more likely to stay in low-energy states. Similarly, it is well known that biologically interesting conformations are often located in regions where the energy function undergoes significant variations, *e.g.*, protein conformations in the transition state. To increase the sampling density in these regions, techniques such as Gaussian sampling [BOvdS99] can be used to sample a pair of conformations and retain a sample with higher probability when the pair exhibits very different energies. Equally important is to identify energy barriers between neighboring nodes in a roadmap while computing transition probabilities. To this end, we may sample the straight-line path between two neighboring nodes and compute the energy of intermediate conformations along the path.

Assume that a good sampling distribution σ can be constructed. If we want to adjust the transition probabilities to account for the non-uniformity of the roadmap so that SRS still converges to the Boltzmann distribution in the limit, one possibility is to define the new transition probability

$$P_{ij} = \begin{cases} \frac{\sigma_i}{d_j \sigma_j} \exp(-\Delta E_{ij}/k_{\mathrm{B}}T) & \text{if } \frac{\varepsilon_j/d_j\sigma_j}{\varepsilon_i/d_i\sigma_i} < 1 \\ \frac{1}{d_i} & \text{otherwise,} \end{cases}$$

where σ_i and σ_j are the probabilities of sampling nodes v_i and v_j. In the synthetic landscape, our empirical results show that the stationary distribution is indeed Boltzmann with this transition probability assignment for non-uniform roadmaps. Work is underway to investigate the effect of this transition probability assignment on other parameters of interest, such as $\mathrm{P_{fold}}$.

It is also important to compare $\mathrm{P_{fold}}$ values obtained by SRS with not just MC simulation, but also Molecular Dynamics as well as with in vitro experiment. For example, the conformations for which $\mathrm{P_{fold}}$ = 0.5 should be in the transition state ensemble, which can be observed with experimental techniques. This would allow us to further validate SRS. Finally, we are also interested in applying SRS to other important questions related to molecular motion, such as the order of formation of secondary structure elements in protein folding.

Acknowledgements This work has been partially funded by an NSF-ITR grants ACI-0082554 and CCR-0086013 and a grant from Stanford's Bio-X program. Apaydın was supported by the D.L. Cheriton Stanford Graduate Fellowship. Brutlag was supported by National Human Genome Research Institute grant HGF02235. Guestrin was supported by a Siebel Scholarship and by the Sloan Foundation. We thank D. Koller, V. Pande and J. Snoeyink for helpful discussions, A. Singh for the ligand-protein modelling software. We also thank the anonymous reviewers for their valuable comments.

References

[ABG$^+$02] M. S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, and J. C. Latombe. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 12–21, 2002.

[ADS02] N.M. Amato, K.A. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 2–11, 2002.

[AGV$^+$02] M. S. Apaydin, C.E. Guestrin, Chris Varma, D.L. Brutlag, and J. C. Latombe. Stochastic roadmap simulation for the study of ligand-protein interactions. In *Bioinformatics*, volume 18, supplement 2, pages 18S–26S, 2002.

[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 & Automation*, pages 932–939, 2001.

[B$^+$77] F.C. Bernstein et al. The protein data bank: A computer-based archival file for macromolecular structure. *J. Mol. Biol.*, 112(3):535–542, 1977.

[BOvdS99] V. Boor, M.H. Overmars, and F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1018–1023, 1999.

[BSA01] O. Burchan Bayazit, Guang Song, and Nancy M. Amato. Ligand binding with obprm and haptic user input. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2001.

[CBA+88] A. Clarke, H. Wilks D. Barstow, T. Atkinson, W. Chia, and J. Holbrook. An investigation of the contribution made by the carboxylate group of an active site histidine-aspartate couple to binding and catalysis in lactate dehydrogenase. *Biochemistry*, 27:1617 – 1622, 1988.

[CV01] C.J. Camacho and S. Vajda. Protein docking along smooth association pathways. *Proc. Nat. Acad. Sci. USA*, 98(19):10636–10641, 2001.

[CWC+86] A. Clarke, D. Wigley, W. Chia, D. Barstow, T. Atkinson, and J. Holbrook. Site-directed mutagenesis reveals the role of a mobile arginine residue in lactate dehydrogenase catalysis. *Nature*, 324:699 – 702, 1986.

[CWHH85] A. Clarke, A. Waldman, K. Hart, and J. Holbrook. The rates of defined changes in protein structure during the catalytic cycle of lactate dehydrogenase. *Biochim. Biophys. Acta*, 829:397 – 407, 1985.

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

[DWH+91] C. Dunn, H. Wilks, D. Halsall, T. Atkinson, A. Clarke, H. Muirhead, and J. Holbrook. Design and synthesis of new enzymes based on the lactate dehydrogenase framework. *Phil. Trans. R. Soc. Lond.*, 332:177 – 184, 1991.

[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.

[Har89] K. Hart. *An investigation of the molecular basis of substrate specificity in lactate dehydrogenase*. Ph.d. thesis, University of Bristol, 1989.

[HCW+87] K. Hart, A. Clarke, D. Wigley, A. Waldman, W. Chiaand D. Barstow, T. Atkinson, J. Jones, and J. Holbrook. A strong carboxylate-arginine interaction is important in substrate orientation and recognition in lactate dehydrogenase. *Biochim. Biophys. Acta*, 914:294 – 298, 1987.

[HLSR75] J. Holbrook, A. Liljas, S. Steindel, and M. Rossmann. Lactate dehydrogenase. *Enzymes*, 11a:191 – 293, 1975.

[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.E. Kavraki, P. Švestka, J. C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. on Robotics & 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.

[MGH+98] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, and A.J. Olson. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.*, 19(14):1639–1662, 1998.

[MRR+53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[RK00] C. Reyes and P. Kollman. Investigating the binding specificity of u1a-rna by computational mutagenesis. *J. Mol. Biol.*, 295(1):1–6, 2000.

[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.

[Saa96] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS, New York, 1996.

[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.

[SH90] K. Sharp and B. Honig. Electrostatic interactions in macromolecules: theory and applications. *Ann Rev Biophys Chem*, 19:301–332, 1990.

[SKS01] J. Shimada, E.L. Kussell, and E.I. Shakhnovich. The folding thermodynamics and kinetics of crambin using an all-atom monte carlo simulation. *J. Mol. Biol.*, 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.

[SP00] M. Shirts and V. Pande. Screen savers of the world, unite! *Science*, 290:1903–1904, 2000.

[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.

[Tea01] IBM Blue Gene Team. Blue gene: A vision for protein science using a petaflop supercomputer. *IBM Systems Journal*, 40(2):310–327, 2001.

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

[TPK02] M. Teodoro, G.N. Jr. Phillips, and L.E. Kavraki. A dimensionality reduction approach to modeling protein flexibility. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 299–308, 2002.

[WHF+88] H. Wilks, K. Hart, R. Feeney, C. Dunn, H. Muirhead, W. Chia, D. Barstow, T. Atkinson, A. Clarke, and J. Holbrook. A specific, highly active malate dehydrogenase by redesign of a lactate dehydrogenase framework. *Science*, 242:1541 – 1544, 1988.

[WKK99] J. Wang, P.A. Kollman, and I.D. Kuntz. Flexible ligand docking: A multiple strategy approach. *Proteins: Structure, Function, and Genetics*, 36(1):1–19, 1999.

A Proof of Lemma 1

Proof: We would like to prove that the distribution π given in (5) is the stationary distribution for the Markov chain induced by the roadmap G. First, note that it is sufficient to show that π satisfies the detailed balance [TK94]:

$$\pi_i P_{ij} = \pi_j P_{ji}, \tag{13}$$

because if (13) holds, then $\sum_j \pi_j P_{ji} = \sum_j \pi_i P_{ij} = \pi_i \sum_i P_{ij} = \pi_i$, as required by the condition for a stationary distribution, given in (2). Now consider two nodes v_i and v_j from the roadmap. Without loss of generality, assume $\frac{\varepsilon_j/d_j}{\varepsilon_i/d_i} < 1$. We have

$$P_{ij} = \frac{1}{d_j}\exp(-\Delta E_{ij}/k_{\mathrm{B}}T) \quad \text{and} \quad P_{ji} = \frac{1}{d_j}.$$

Substituting these expressions into (13), we can easily verify that (13) is satisfied, after some simplification.

□

B Theorem 1

Let S be any subset of the conformation space $\mathcal{C}$ with relative volume $\mu(S) > 0$. 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 $\beta(S)$ and the estimate $\pi(S)$ from the roadmap is given by

$$(1-\delta)\beta(S) - \varepsilon \le \pi(S) \le (1+\delta)\beta(S) + \varepsilon, \tag{14}$$

with probability at least $1-\gamma$.

Furthermore, if $\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S \ge 1$, then the number of roadmap nodes N required is given by

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

$$\left.\frac{4}{\left[\sqrt{(\mu(S)+\varepsilon)^2+4\varepsilon\alpha(\mathcal{C})}-(\mu(S)+\varepsilon)\right]^2},\right.$$
$$\frac{[\alpha(\mathcal{C})+\mu(S)(\delta+1)]^3}{2\alpha(\mathcal{C})^2\mu(S)^3\delta^2\left[\alpha(\mathcal{C})+\mu(S)(\delta+1)+\alpha(\mathcal{C})\delta\right]},$$
$$\left.\frac{[\alpha(\mathcal{C})+\mu(S)(\delta+1)]^2}{\alpha(\mathcal{C})^2\mu(S)^2\delta^2}\right\}.$$

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

Proof:

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

Lemma 2 (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:*

$$\begin{aligned} &P(\overline{Y} - E[Y] \geq \varepsilon) \leq e^{-\frac{2n\varepsilon^2}{(b-a)^2}}, \quad \textit{and} \\ &P(E[Y] - \overline{Y} \geq \varepsilon) \leq e^{-\frac{2n\varepsilon^2}{(b-a)^2}}.\square \end{aligned} \tag{15}$$

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 S:

Lemma 3 *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 S, then:*

$$\mu(S) - \varepsilon_1 \leq \frac{K}{N} \leq \mu(S) + \varepsilon_1; \tag{16}$$

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

Proof: Application of Hoeffding's inequality, where the random variable Y is the indicator that a point falls in the set S. By the law of large numbers, $E[Y] = \mu(S)/\mu(\mathcal{C}) = \mu(S)$. The empirical mean $\overline{Y} = K/N$ and Y is an indicator, thus, $Y \in [0,1]$. The proof is concluded by applying Lemma 2. $\square$

We would like to have, with high probability, at least one milestone in the 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 3, we know that $K \geq \lfloor N(\mu(S) - \varepsilon_1) \rfloor$. Thus:

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

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

$$\alpha(\mathcal{F}) - \varepsilon \cdot \mu(\mathcal{F}) \leq \frac{\mu(\mathcal{F})}{M} \sum_i \exp(-E(Y_i)/k_{\mathrm{B}}T) \leq \alpha(\mathcal{F}) + \varepsilon \cdot \mu(\mathcal{F}); \quad (17)$$

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

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

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

We will apply Lemma 4 twice, first for computing the Boltzmann integral in the set S, obtaining the bound:

$$\alpha(S) - \varepsilon_2\mu(S) \leq \frac{\mu(S)}{K}\sum_{i\in S}\exp(-E(Y_i)/k_{\mathrm{B}}T) \leq \alpha(S) + \varepsilon_2\mu(S); \tag{18}$$

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

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

The second bound concerns the integral over the whole space:

$$\alpha(\mathcal{C}) - \varepsilon_3 \leq \frac{1}{N}\sum_{j}\exp(-E(Y_j)/k_{\mathrm{B}}T) \leq \alpha(\mathcal{C}) + \varepsilon_3; \tag{19}$$

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

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

In the remainder of this proof, we will assume that equations (16), (18) and (19) 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 1 the stationary distribution on the roadmap can be rewritten as:

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

Applying the bound on Equation (16) we get:

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

$$\leq \pi(S) \leq$$

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

rearranging:

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

$$\leq \pi(S) \leq$$

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

We can now apply the bounds in Equations (18) and (19):

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

This expression can be rewritten as:

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

which finally leads us to the statement of our theorem:

$$(1-\delta)\beta(S)-\varepsilon \leq \pi(S) \leq (1+\delta)\beta(S)+\varepsilon;$$

where ε and δ impose the following constraints:

$$\varepsilon \quad\geq\quad \frac{\varepsilon_2(\mu(S)+\varepsilon_1)}{\alpha(\mathcal{C})-\varepsilon_3}; \tag{20}$$

$$\delta \quad\geq\quad \frac{\varepsilon_1\alpha(\mathcal{C})+\varepsilon_3\mu(S)}{\mu(S)\,(\alpha(\mathcal{C})-\varepsilon_3)}. \tag{21}$$

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

$$N \quad\geq\quad \frac{\ln(2/\gamma_1)}{2\varepsilon_1^2}; \tag{22}$$

$$N \quad\geq\quad \frac{\ln(2/\gamma_2)\|\exp(-E(v)/k_\mathrm{B}T)\|_S^2}{2(\mu(S)+\varepsilon_1)\varepsilon_2^2}; \tag{23}$$

$$N \quad\geq\quad \frac{\ln(2/\gamma_3)\|\exp(-E(v)/k_\mathrm{B}T)\|_S^2}{2\varepsilon_3^2}; \tag{24}$$

$$\gamma \quad\geq\quad \gamma_1+\gamma_2+\gamma_3. \tag{25}$$

Given any $\varepsilon > 0$, $\delta > 0$ and $\gamma > 0$, we can use constraints (20) — (25) 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: $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \tilde{\varepsilon}$ and $\gamma_1 = \gamma_2 = \gamma_3 = \gamma/3$.

Let's first consider the ε constraint on Equation (20), which can now be written as:

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

Rearranging, we have that:

$$0 \leq \varepsilon\alpha(\mathcal{C}) - \tilde{\varepsilon}^2 - \tilde{\varepsilon}(\mu(S) + \varepsilon).$$

Solving for $\tilde{\varepsilon}$, we obtain:

$$\tilde{\varepsilon} \leq \frac{\sqrt{(\mu(S) + \varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})} - (\mu(S) + \varepsilon)}{2} \tag{26}$$

Using a similar manipulation of the δ constraint on Equation (21), we can write:

$$\tilde{\varepsilon} \leq \frac{\alpha(\mathcal{C})\mu(S)\delta}{\alpha(\mathcal{C}) + \mu(S)(\delta + 1)}. \tag{27}$$

We can now consider the constraints on N given by Equations (22) — (24). Note that for the case of $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \tilde{\varepsilon}$ and $\gamma_1 = \gamma_2 = \gamma_3$, only the constraints in Equation (23) and Equation (24) will be binding, assuming $\| \exp(-E(v)/k_{\mathrm{B}}T)\|_S \geq 1$, i.e. the range of Boltzmann ratio is greater than 1 in S. These constraints can now be written as:

$$\begin{aligned} N \geq \quad \max \quad & \left\{ \frac{\ln(6/\gamma)\| \exp(-E(v)/k_{\mathrm{B}}T)\|_S^2}{2(\mu(S) + \tilde{\varepsilon})\tilde{\varepsilon}^2}, \right. \\ & \left. \frac{\ln(6/\gamma)\| \exp(-E(v)/k_{\mathrm{B}}T)\|_S^2}{\tilde{\varepsilon}^2} \right\}. \end{aligned}$$

Substituting the constraints on $\tilde{\varepsilon}$ given by Equations (26) and (27), we can obtain the value of N:

$$\begin{aligned} N = \quad & \ln(6/\gamma)\| \exp(-E(v)/k_{\mathrm{B}}T)\|_S^2 \cdot \\ & \max \left\{ \quad \frac{4}{\left[(\mu(S) - \varepsilon) + \sqrt{(\mu(S) + \varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})}\right] \left[\sqrt{(\mu(S) + \varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})} - (\mu(S) + \varepsilon)\right]^2}, \right. \end{aligned}$$

$$\frac{4}{\left[\sqrt{(\mu(S)+\varepsilon)^2+4\varepsilon\alpha(\mathcal{C})}-(\mu(S)+\varepsilon)\right]^2},$$

$$\frac{[\alpha(\mathcal{C})+\mu(S)(\delta+1)]^3}{2\alpha(\mathcal{C})^2\mu(S)^3\delta^2\left[\alpha(\mathcal{C})+\mu(S)(\delta+1)+\alpha(\mathcal{C})\delta\right]},$$

$$\left.\frac{[\alpha(\mathcal{C})+\mu(S)(\delta+1)]^2}{\alpha(\mathcal{C})^2\mu(S)^2\delta^2}\right\}.$$

$\square$

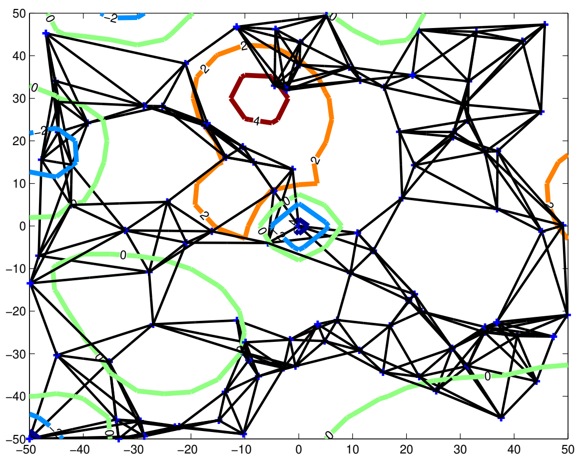


Figure 1. A stochastic conformational roadmap superimposed on the contour plot of a fictitious energy landscape on a 2-D space.

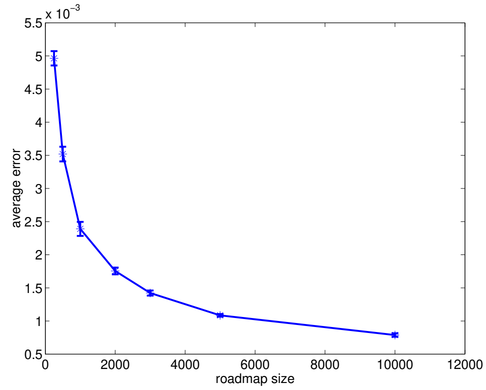
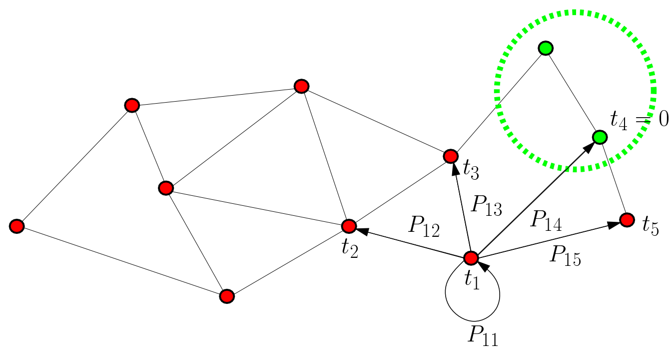


Figure 2. Average error in SRS estimates of the stationary distribution.



$$t_1 = 1 + P_{11} \cdot t_1 + P_{12} \cdot t_2 + P_{13} \cdot t_3 + P_{15} \cdot t_5$$

Figure 3. Illustration for first-step analysis.

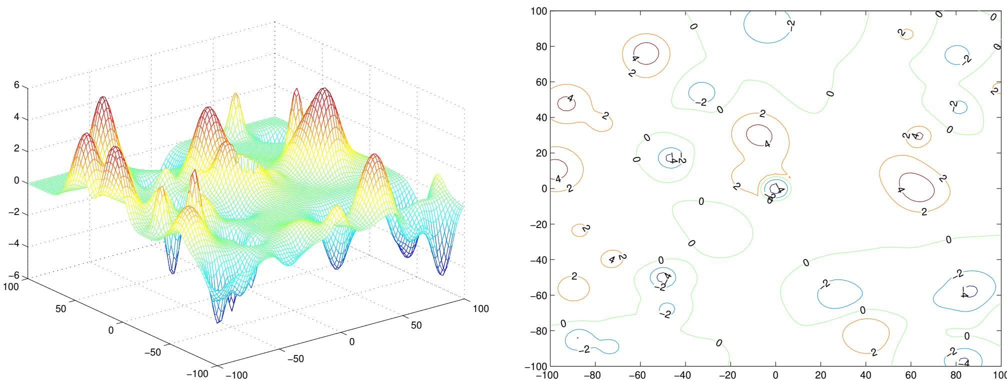


Figure 4. The 2-D synthetic energy landscape used in our study, along with its contour plot.

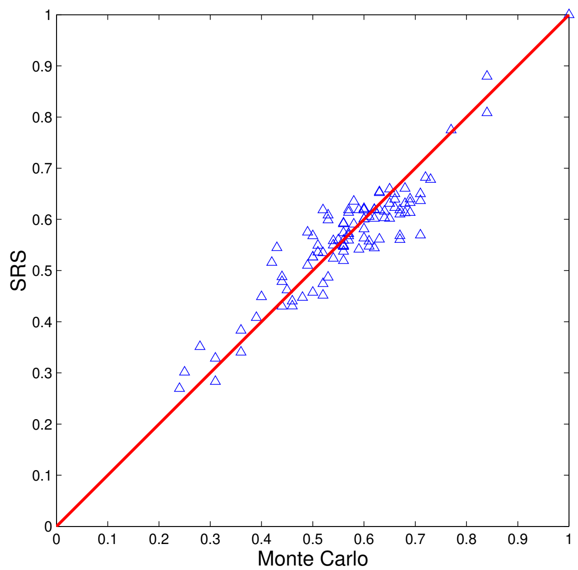


Figure 5. The correlation of P_{fold} values computed by MC simulation and SRS on a fictitious energy function.

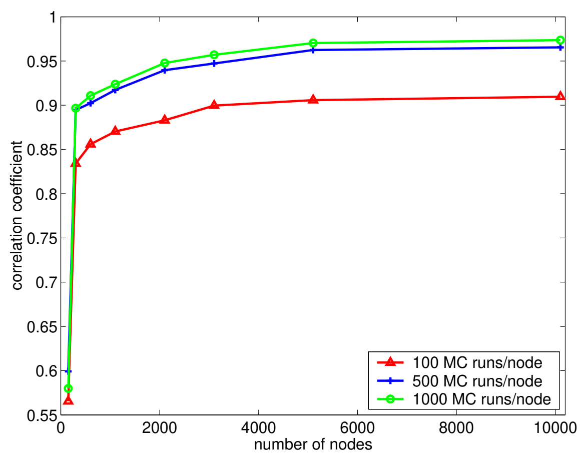


Figure 6. Correlation coefficient κ as a function of the number of nodes in the roadmap.

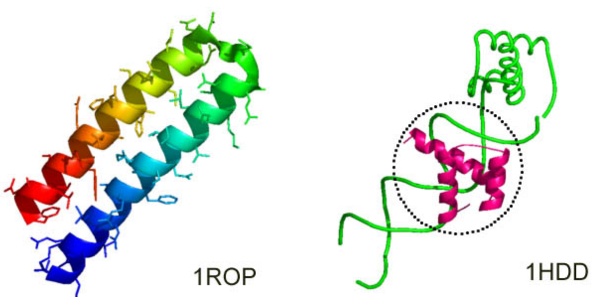


Figure 7. Two proteins used in our study: 1ROP (one monomer) and 1HDD (circled) in complex with DNA.

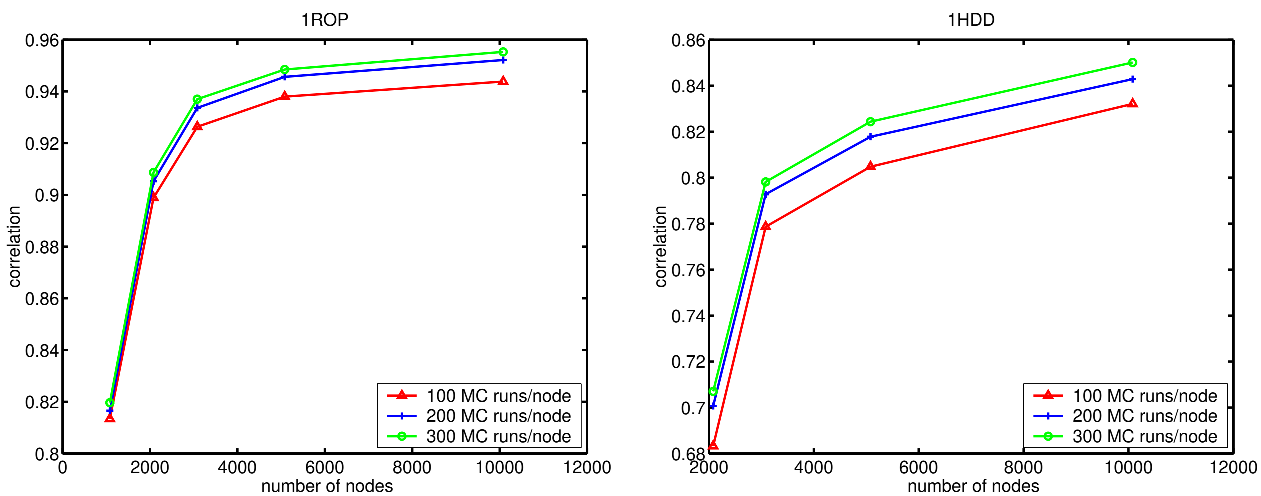


Figure 8. The correlation of $P_{\rm fold}$ values for 1ROP and 1HDD, computed by SRS and MC simulation.

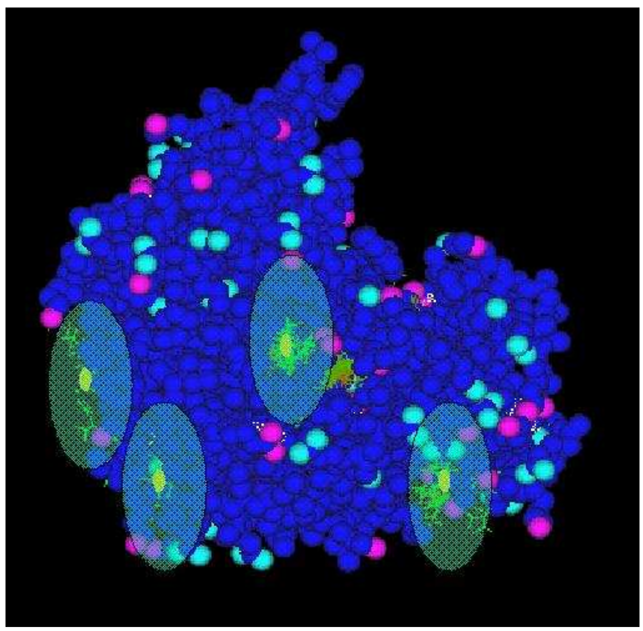


Figure 9. Funnels of attraction of four potential binding sites on lactate dehydrogenase.

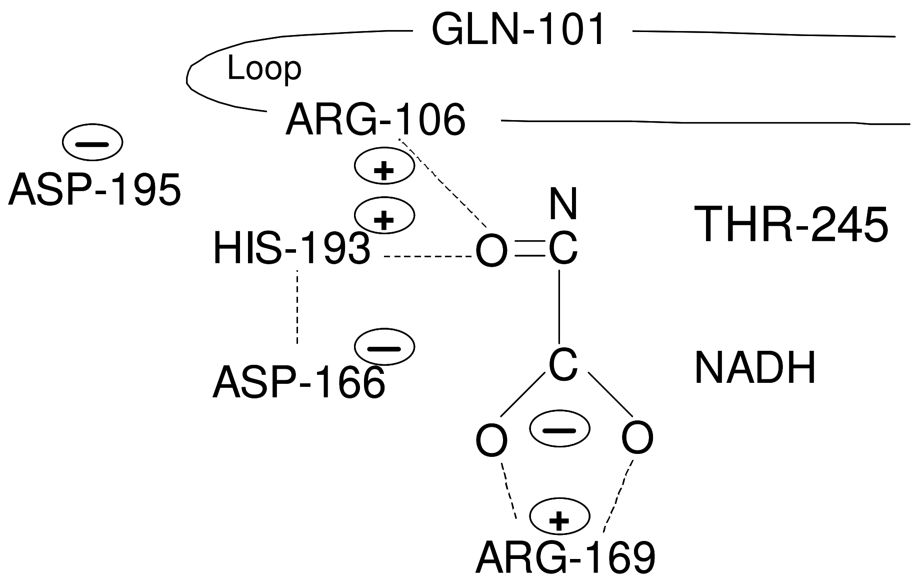


Figure 10. The chemical environment of LDH-NADH-substrate complex. Hydrogen atoms are not explicitly modeled.

Mutant	Bound Energy (kcal/mol)	Escape Time	Expected Effect
Wild type	0.233467	2.1e+06	N/A
His193 → Ala and Arg106 → Ala	4.526738	7.7e+03	Decrease in escape time.
His193 → Ala	-1.370748	4.6e+04	Decrease in escape time.
Arg106 → Ala	1.305369	7.2e+03	Decrease in escape time.
Asp195 → Asn	-9.258782	1.1e+07	Increase in escape time.
Gln101 → Arg	-8.516694	1.4e+06	No effect
Thr245 → Gly	-6.628186	1.8e+05	Decrease in escape time.

Table 1. Effects of mutations on the catalytic site.

Table 2. Ligand-protein complexes used in the experiments and the number of DoFs.

Protein	Ligand	dofs
1LDM	oxamate	7
1A05	3-isopropylmalate	10
3TPI	Ile-Val	13
4TS1	hydroxylamine	9
1CJW	COA-S-ACETYL tryptamine	21
1AID	THK UCSF8	14
1STP	streptavidin	11

Table 3. Escape times from binding sites.

	Binding Sites				
Protein	Active	1	2	3	4
1LDM	5.8e+06	1.6e+07	1.1e+06	3.7e+06	4.5e+05
1AO5	4.1e+10	1.2e+07	7.9e+06	1.2e+05	2.9e+04
3TPI	1.0e+10	1.1e+06	1.8e+05	1.0e+05	6.6e+05
4TS1	2.4e+10	5.4e+06	4.2e+07	7.2e+05	2.2e+06
1CJW	6.3e+06	8.2e+06	5.6e+05	1.5e+05	1.9e+05
1AID	1.4e+06	2.8e+07	5.0e+05	1.2e+05	2.1e+06
1STP	7.0e+08	6.4e+06	2.2e+06	8.5e+05	2.0e+06