Roadmap Methods for Protein Folding

Mark Moll, David Schwarz, Lydia E. Kavraki

Abstract—*Protein folding refers to the process whereby a protein assumes its intricate three-dimensional shape. This chapter reviews a class of methods for studying the folding process called roadmap methods. The goal of these methods is not to predict the folded structure of a protein, but rather to analyze the folding kinetics. It is assumed that the folded state is known. Roadmap methods build a graph representation of sampled conformations. By analyzing this graph one can predict structure formation order, the probability of folding, and get a coarse view of the energy landscape.*

Keywords: protein folding, folding kinetics, roadmap methods, conformation sampling techniques, energy landscape.

1 Introduction

Protein folding refers to the process whereby a protein assumes its intricate three-dimensional shape. Different aspects of this problem have attracted much attention in the last decade. Both experimental and computational methods have been used to study protein folding and there has been considerable progress [1–7]

This chapter reviews a class of methods for studying protein folding called *roadmap methods* [8–19]. These methods are relatively new and are still under active development. Roadmap methods are computational methods that have been developed to understand the process or the mechanism by which a protein folds or unfolds. It is typically assumed that the folded state is already known. Note that this is not a comprehensive survey of all existing computational protein folding methods. In particular, it does not cover Molecular Dynamics (MD) methods [20], Monte Carlo methods (MC) [21], the use of coarse grain models in simulations and many others.

Many papers (see for example [20–22]) have discussed the advantages and disadvantages of traditional computational methods for studying protein folding. Some of the drawbacks include the fact that classical MD/MC simulations typically compute only one trajectory, that escaping local minima can be very difficult and that the process has no memory to recognize whether conformations have been visited in the past or not. These issues led some researchers to develop enhanced versions of MC and MD, which take advantage of laboratory data, non-uniform or accelerated timescales, modified energy functions, parallelism, biases away from previously generated conformations, and other modifications (for examples, see [23–26]). Other researchers, inspired by advancements in robot modeling and by the need for alternative protein modeling methods, began to build so-called roadmaps to explore the conformational space of proteins. A roadmap is a representation of many conformations and the transitions between them as a graph data structure. Roadmap-based methods were originally developed in robotics [27] where the configuration (conformation) space of a robot is explored in order to find a collision-free path that will take the robot from an initial position to a final position. By taking advantage of the analogy between robots and molecules, in which the main molecular chain of a protein corresponds to an articulated robot, roadmap methods were adapted to study how a protein can attain a known final shape. Roadmap methods were significantly modified and enhanced to address the folding problem. Their application to the folding problem is still relatively new and not as well-understood as MD/MC simulations. They seem to offer vast computational improvements and potentially increased coverage of the conformational space compared to traditional methods. This could mean that 'interesting' areas of the conformational space can quickly be discovered, and—if necessary—further explored with traditional methods. Yet, it is not clear how much (if anything) is lost by the use of coarse approximations. This chapter surveys some of the most promising roadmap methods for protein folding [9–19].

2 Background

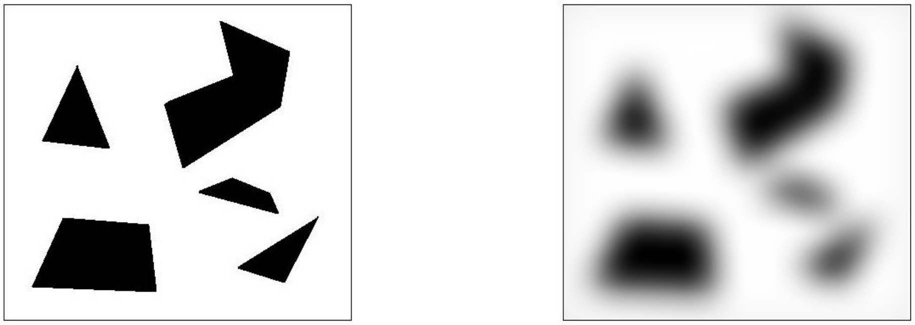
2.1 Protein Representation

The simplest representation of a protein is a vector that contains the Cartesian coordinates of all atoms in a conformation. This is the representation used in MD/MC simulations; molecular potential energy functions are almost always parameterized by atomic coordinates in Cartesian space (e.g., [28]).

The drastic changes in the conformation of a protein occur, however, with rotations about certain bonds. Often, a vector of bond rotations is used as a more compact representation of a protein. The amount of rotation about a single bond relative to some reference state is called the dihedral angle. This representation ignores the stretching of bond lengths and bond angles, but these effects are often negligible compared to the bond rotations. Efficient ways to calculate the Cartesian coordinates of all atoms given the dihedral angles of a protein are given in [29].

Another way to represent a protein is to model flexibility at the level of secondary structure. A molecule is divided into α-helices, β-sheets, and connecting loops. The sequence of secondary structure elements is represented by a sequence of vectors. Rotational degrees of freedom are assigned at the junctions where the vectors meet. The α-helices and β-sheets can twist about their axis, and the loop regions are allowed to extend in the direction of their vector. In this representation, traditional energy functions cannot be used, but it is possible to approximate molecular energy using a simple potential function [30].

In roadmap methods for protein folding, all of the above representations have been used, but the most popular one is the representation of conformations by dihedral angles. As will be explained in the next section, roadmaps sample the conformation space of a protein. The dihedral angle representation of a protein readily allows the generation of samples that have properties desirable for roadmap-based methods.



(a) A two-dimensional robotic configuration space. Black shapes represent sets of configurations that place the robot in collision with obstacles.

(b) A two-dimensional molecular conformation space, which could correspond to a molecule with two rotatable bonds. White regions are low-energy, black high-energy, and gray intermediate-energy. The higher the energy of a conformation, the less likely a molecule is to assume that conformation.

Figure 1: Robotic configuration space vs. molecular conformation space.

2.2 Roadmap Algorithms for Robot Motion Planning

The idea of using a roadmap to represent properties of a complex space originated in robotic motion planning [27, 31]. In motion planning, a collision-free path between a start and goal configuration of a robot is computed. Consider a long articulated robot for the moment. The degrees of freedom of such a robot correspond to moving its joints. The set of all configurations of a robot is called its configuration space. Each point in this space corresponds to a robot configuration. A simple, two-dimensional robotic configuration space is illustrated in Figure 1(a). The subset of configurations where the robot does not collide with any obstacles (including the robot itself) is called the free space, and is drawn white in Figure 1(a). The set of configurations in which the robot collides with itself or a workspace obstacle is called the occupied space and is drawn in black in Figure 1(a). Motion planning can thus be phrased as the problem of finding a curve (a path) that lies completely in the free part of the configuration space.

Computing the free space *exactly* is a very hard problem. The size of the configuration space and the complexity of the motion planning problem grow exponentially with the number of degrees of freedom [32]. Sampling based techniques called Probabilistic Roadmap Methods (PRMs) [27] build a roadmap: a graph representation of the free space, where nodes corresponds to configurations and edges to paths between them. This roadmap is computed as follows. First, a large number of collision-free configurations are sampled. Next, for each configuration, an attempt is made to find a path to some of its nearest neighbors. These local paths can simply be straight lines in the configuration space. If the path between two configurations lies entirely in the free space, it is added to the roadmap. The motion planning problem is now easily solved. The start and goal configuration are connected to their nearest neighbors in the roadmap. The path is then found by performing a simple graph search to connect the start to the goal. Note that the roadmap has to be computed only once for a given robot, and that many motion planning queries can be solved with the same roadmap. PRMs are able to solve motion planning problems in very high-dimensional configuration spaces, but they do not guarantee completeness, *i.e.*, they do not always find a path if one exists. Instead, they have been shown to be *probabilistically complete*, *i.e.*, if a path exists, then with high probability the PRM algorithm will find it. This probability goes to one as the number of sampled configurations increases. Many variations of the basic PRM algorithm have been proposed to increase the sampling of configurations in difficult areas (such as narrow passages). A discussion of the PRM algorithm and its variations can be found in [31].

For certain applications it is known *a priori* that only one motion planning query will need to be solved, so sampling the entire configuration space may be unnecessary. This observation lead to a different class of sampling-based path planning algorithms in which a tree of configurations is grown from the start to the goal configuration and/or vice versa. The three main variations within this class are called Rapidly-exploring Random Trees (RRTs) [33], Expansive Spaces Trees (ESTs) [34], and Path-Directed Subdivision Trees (PDSTs) [35]. RRTs grow a tree of configurations as follows. First, a random configuration, which may be in collision, is sampled. Next, the nearest configuration in the existing tree to the one just sampled is found. Initially, the tree consists of just the start configuration. From the nearest configuration, a new configuration is found some distance in the direction of the randomly sampled configuration. This process is repeated until the tree is close to the goal configuration. This algorithm tends to 'pull' the tree growth in the direction of unexplored parts of the configuration space. ESTs, on the other hand, can be thought of as 'pushing' the tree growth in promising areas. During each iteration of the EST algorithm, a previously sampled configuration is selected at random and a new configuration is sampled in a neighborhood of it. The key in the algorithm is the probability distribution function used to sample the previous configurations. The EST assigns a probability to each configuration that is proportional to the distance to the *k* nearest neighbors and inversely proportional to the number of times the configuration has been selected before. Sampling using this distribution expands the trees towards unexplored areas of the configuration space. PDSTs represent the trees somewhat differently from other tree-based and roadmap methods. Rather than maintaining a set of nodes and edges, a PDST consists of a set of edges, representing paths, joined at branches. It also maintains a cell decomposition of the configuration space and assigns paths to cells. At each step of the PDST exploration, an edge is selected based on an estimate of how well the area around each edge has already been explored (measured using the cell decomposition), and a new edge is created starting from a random point along the selected edge. In this way, the tree expands outward from its origin and the updating of the cell decomposition leads the expansion of the tree to less well-sampled areas.

Both roadmap and tree-based path planning and exploration algorithms have been used to study the dynamic properties of proteins, including their folding behavior but also their interactions with other molecules [36–40]. In order to apply these robotics-based methods to complex molecular systems, however, some adaptations of the algorithms are necessary, as will be presented in the following sections.

3 Roadmaps for Protein Folding

Conceptually, there is an analogy between high-energy areas in the conformation space of a molecular system and obstacles, and between low-energy areas and free space (see figure 1(b)). There may not be a single cut-off energy threshold, however, to separate the conformation space in black and white regions. Molecular conformation spaces therefore have a fuzzier notion of collision and free space than robotic configuration spaces, as is shown in Figure 1(b), and there are other important differences between exploring the free space of a robot and the free space of a biomolecule. In a biochemical context, low-energy paths are of specific interest, rather than paths in general. In folding, in particular, if it is assumed that the folded state of a protein is known, then researchers would like to find how the protein unfolds and refolds and determine some aggregate properties of these pathways, such as the overall folding rate and probability of any given structure to proceed to a folded state. It is important to note that the goal of roadmap methods is not to predict the folded state from a sequence of amino acids. The interest is in folding kinetics: the aim is to get a better understanding of *the process or mechanism* by which a protein folds and unfolds. It is assumed that the folded state has already been determined.

The essential ingredients of any roadmap method are the choice of degrees of freedom, the conformation sampling technique, and the way to connect conformations to form a roadmap. Another important ingredient for roadmaps of molecular systems is the energy model. So far simplified energy models have been used. It remains to be seen how accurate these models are for complex problems. This section will review how roadmap based methods can provide new insights into folding kinetics.

Before getting into the details of specific methods, it is worth mentioning that the idea of using roadmap methods to study problems in molecular biology originated with Singh *et al.* [41], who adapted the PRM algorithm to study the docking of a ligand to a protein. Nodes in the roadmap represented conformations and poses of the ligand, and were sampled at random around the protein and kept or rejected based on their energy. Neighboring nodes were connected with an edge if a set of conformations sampled on a straight line in configuration space between them were all below an energy cutoff, and edges were labeled with transition probabilities depending on the energy difference between the nodes at either end. This work permitted the identification of active sites in proteins.

Several research groups extended and adapted this work, refocusing it on protein folding mechanisms [9–19]. The general trends of this ongoing research include tweaking the energy function, edge weights and/or node sampling schemes. The goal of such work is ultimately to develop methods in which the final energy distribution of the set of nodes and paths in the roadmap corresponds to the energy distribution predicted by statistical mechanics (Boltzmann-like). Given a high-quality roadmap, it should be possible to determine properties of the protein's motion and folding behavior from all-path analyses.

In general, the folding kinetics can be analyzed by looking at many paths in the roadmap. There are three fundamentally different ways to construct and interpret the roadmap. In the first method (described in section 3.1), the object is to compute the most energetically favorable paths between the folded state and denatured states and to consider those the folding pathways. This is the approach taken by Amato *et al.* [9–13]. In the second method, the weights of edges in a roadmap are interpreted as probabilities and the roadmap gives rise to a Markov chain. The folding pathways are analyzed by performing random walks on the roadmap or by computing the limit distribution from the matrix of state transition probabilities. This is the approach taken by Apaydin *et al.* [16–19] and is described in section 3.2. Finally, in section 3.3, we describe the third method, proposed by Singhal *et al.* [14,15], which combines roadmap methods with MD/MC methods.

3.1 PRMs for Protein-Folding Pathways

In the work of Amato *et al.* [9–13] the backbone ϕ and ψ dihedral angles are taken to be the degrees of freedom. The side chains are assumed to be rigidly attached to the backbone. For a protein consisting of n residues there are $2(n-1)$ degrees of freedom (the first and last rotational angle do not contribute). Conformations can be sampled by randomly picking angles from the allowable range. The sampling can be based on Ramachandran plots [42], but this approach has a very small probability of producing of conformations without steric clashes. In early work, Amato *et al.* [13] used Gaussian sampling around the folded state with various standard deviations to create new conformations. This works well for proteins with approximately 60 residues, but it still does not scale up to larger proteins with over 100 residues. A more successful strategy is the following: Instead of sampling only around the native state, conformations are sampled around all previously sampled conformations. This is done in a way that creates a 'wavefront' of conformations growing outwards from the native state. The conformations are partitioned into bins based on the number of native contacts. A *native contact* is defined as a pair of C_α atoms that are within 7 Å of each other in the native state. The bins are equal-sized and the number of bins is proportional to the number of native contacts in the native state. A conformation q is accepted based on its energy $E(q)$. When a structure is generated, it is checked for collision of side chains, and rejected if any are found. If it passes that test, the energy consists of a term favoring documented secondary structure via known backbone hydrogen and disulfide bonds, and a term for hydrophobic interactions.

The probability of accepting a conformation q is:

$$P(\text{accept } q) = \begin{cases} 1 & \text{if } E(q) < E_{\min}, \\ \frac{E_{\max}-E(q)}{E_{\max}-E_{\min}} & \text{if } E_{\min} \leq E(q) \leq E_{\max}, \\ 0 & \text{if } E(q) > E_{\max}. \end{cases}$$

Thus, all low-energy conformations are kept, as well as some of the medium-energy conformations, in order to connect the low-energy areas. The energy thresholds $E_{\min}$ and $E_{\max}$ are set at 50,000 kJ and 89,000 kJ, respectively. The accepted conformations are put in the appropriate bin. The sampling process iteratively tries to fill all bins, starting with the bin with 100% native contacts. Once a neighboring bin has at least n conformations, sampling is performed around conformations in that bin, in order to fill the succeeding bins. Although this sampling method does not seem to correspond to a Boltzmann distribution of states, it still may capture some of the essential folding properties such as contact formation order [11].

The second phase in the roadmap construction is the connection of the sampled conformations. For each conformation the method attempts to connect each node to its k nearest neighbors. The ϕ and ψ angles are linearly interpolated and energy is checked along the line in conformation space connecting a conformation q_0 and one of its neighbors q_1. If the energy does not exceed some threshold, the edge connecting q_0 and q_1 is added to the roadmap.

The edge is given a weight that depends on the energy along the line connecting q_0 and q_1. Suppose the energy of the sequence of conformations $q_0 = c_0, c_1, c_2, \dots, c_{n-1}, c_n = q_1$ along the line connecting q_0 and q_1 has been computed. The probability of moving from c_i to c_{i+i} is

$$P_i = \begin{cases} e^{\frac{-\Delta E_i}{kT}} & \text{if } \Delta E_i > 0, \\ 1 & \text{if } \Delta E_i \leq 0. \end{cases}$$

Here, $\Delta E_i = E(c_{i+1}) - E(c_i)$. The weight of the edge between q_0 and q_1 is then defined as

$$w(q_0, q_1) = \sum_{i=0}^{n-1} - \log P_i.$$

The edge weight is intended to encode the likelihood of going from one conformation to another given the energy profile of the path.

After the roadmap is constructed, the folding pathways can be extracted. Starting from the native structure, the shortest path to every other conformation can be found using Dijkstra's algorithm [43].

This roadmap construction method was tested on 14 proteins with 56 to 110 residues, including Protein G and Protein A [11]. Roadmaps were constructed in 2–15 hours. From this, many folding pathways can be extracted and their properties analyzed. Of particular interest is the order of secondary structure formation along each path between the stable unfolded states and the folded state. This order provides a rough overview of the folding mechanism of the protein, and can often be determined by laboratory experiment, thereby providing a criterion by which to validate the roadmap method.

Using a constructed roadmap, the order of secondary structure formation for a single path from an unfolded to folded state is determined by, for each native contact in a secondary structure element, finding the first conformation along the path that contains that contact. Along a single path, the appearance time for a secondary structure element is computed as the mean of the appearance times for all of its contacts. Overall, the predicted secondary structure formation order is the order with the greatest frequency over all paths. For the experimental set of 14 proteins, this analysis of the roadmap correctly predicted the formation order of secondary structure in all cases where laboratory data was available for comparison.

In later work [10], the same group that developed the original methods did a more detailed study of Proteins L and G. These proteins both consist of an α-helix and one four-stranded β-sheet. In spite of this structural similarity, the secondary structures are experimentally documented to form in different orders. PRM analysis correctly predicted these differences in secondary structure formation order.

In their latest work [9], Thomas *et al.* noticed that even the bin-based construction method described above often requires 10,000 or more samples to construct a complete roadmap for relatively small (60-100 residue) proteins. For more typical protein sizes, this poor scaling rapidly becomes prohibitive. As a result, Thomas *et al.* [9] developed a new sampling method based on rigidity analysis of each sampled conformation. Using information about constraints on motion such as disulfide bridges and hydrogen bonds, this analysis classifies each bond as *independently flexible*, *dependently flexible*, or *rigid*. Independently flexible bonds may be rotated without any effect on other degrees of freedom. Dependently rotatable bonds may rotate but necessarily cause other related bonds to rotate also. Rigid bonds, as the name suggests, generally cannot rotate because they are part of a fully-constrained cluster of atoms. Dependently flexible bonds form sets with fewer than the expected number of degrees of freedom.

Under rigidity-based sampling, new samples are generated by perturbing the dihedral angles of existing conformations in a non-uniform way. Specifically, independently flexible bonds are rotated with a high probability, P_{flex}. Rigid bonds are rotated with a low but non-zero probability P_{rigid}. For sets of dependently flexible bonds with k internal degrees of freedom, k are selected at random and rotated with probability P_{flex}, and the remaining bonds are rotated with probability P_{rigid}. Thomas *et al.* found that allowing rigid bonds to rotate helps the method attain better coverage of the conformation space, while biasing rotations to occur most often for flexible bonds focuses the sampling on regions of the conformation space most likely to be accessible to a real protein.

When tested on a set of 26 proteins, it is reported [9] that rigidity-based sampling yielded roadmaps with substantially better connectivity (measured as edges per node) than earlier sampling methods. In many cases, this could often be accomplished using a quarter to half as many nodes as were necessary to produce the roadmap under Gaussian sampling. In addition to correctly predicting the secondary structure formation order of Proteins G and L, analysis of roadmaps created using rigidity sampling also correctly predicted the order of secondary structure formation of NuG1 and NuG2. PRM analysis by Thomas *et al.* without rigidity sampling had previously failed to predict the order of structure formation in these proteins.

3.2 Stochastic Roadmap Simulation

Stochastic Roadmap Simulation (SRS), developed by Apaydin *et al.* [16–19] is a general technique to study molecular motion. The method derived its early inspiration from the work of Singh *et al.* [41], who were attempting to find a way to predict active sites in proteins using roadmap methods.

The roadmap construction in SRS is straightforward. First, a number of conformations is sampled independently at random from the conformation space. Each conformation is connected to its k nearest neighbors. The transition probability P_{ij} of an edge connecting nodes v_i and v_j is defined as

$$P_{ij} = \begin{cases} \frac{1}{d_j} e^{\frac{-\Delta E_{ij}}{k_B T}} & \text{if } \frac{\varepsilon_j / d_j}{\varepsilon_i / d_i} < 1, \\ \frac{1}{d_i} & \text{otherwise,} \end{cases}$$

where ε_i and ε_j are the Boltzmann factors for conformations c_i and c_j, and d_i and d_j are the number of neighbors for v_i and v_j. The Boltzmann factor of a conformation c is defined as $\varepsilon = \exp(-E(c)/k_b T)$. A self-transition is added with probability $P_{ii} = 1 - \sum_{i \neq j} P_{ij}$, so that all transition probabilities of a node add up to 1. The energy $E(c)$ is a hydrophobic-polar (H-P) energy function [30], in which each amino acid residue is classified as hydrophobic or polar, and favorable energy is computed for hydrophobic residues in contact with (within a cutoff distance of) each other. Conformations are also checked for steric clashes (overlapping atoms), and rejected if necessary.

A random walk on this roadmap is defined as follows. Starting at node v_i, a neighbor v_j is chosen uniformly at random. A move from v_i to v_j is accepted with probability

$$A_{ij} = \begin{cases} \frac{d_i}{d_j} e^{\frac{-\Delta E_{ij}}{k_B T}} & \text{if } \frac{\varepsilon_j / d_j}{\varepsilon_i / d_i} < 1, \\ 1 & \text{otherwise.} \end{cases}$$

Each neighbor of v_i has a probability of $1/d_i$ of being chosen. So the probability of a transition from v_i to v_j is $\frac{1}{d_i} A_{ij} = P_{ij}$.

If a random walk is made on this roadmap, then each state i has a probability π_i of being visited. As a random walk continues for an infinitely long time, assuming the Markov chain is ergodic, the probabilities π_i converge to fixed values that are the same for *any* random walk. Moreover, if the conformation space is sampled more and more finely, it can be shown that the limit distribution of the roadmap is the same as the limit distribution of an MC simulation [18]. In other words, the resulting distribution is theoretically consistent with the Boltzmann distribution of energies predicted by statistical mechanics, and, equivalently, with the results of a large number of Monte Carlo simulations.

Once constructed, the roadmap can be interpreted as a Markov chain, and therefore be analyzed using techniques from Markov-chain theory. This can be used to calculate a quantity for each node called P_{fold}, the probability that the structure at that node will become completely folded before it becomes completely unfolded. This quantity can be used to estimate which structures constitute the transition state of the folding process, as well as to estimate the folding time for the protein.

Let $\mathcal{F}$ denote the set of nodes that correspond to conformations that are considered folded. Now suppose there is another stable state called the unfolded state. Let $\mathcal{U}$ denote the set of nodes corresponding to conformations close to the unfolded state. The probability of folding, P_{fold}, also called the transmission coefficient [44], for a given node v_i can be written as

$$P_{\text{fold}}^{(i)} = \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 1 + \sum_{v_j \in \mathcal{U}} P_{ij} \cdot 0 + \sum_{v_j \notin (\mathcal{F} \cup \mathcal{U})} P_{ij} \cdot P_{\text{fold}}^{(j)}.$$

The probability of folding is conditional on the first transition. If a node in $\mathcal{F}$ is reached, then $\mathcal{F}$ has been reached before $\mathcal{U}$ with probability 1. Similarly, if a node in $\mathcal{U}$ is reached, then $\mathcal{F}$ has been reached before $\mathcal{U}$ with probability 0. Otherwise, $P_{\text{fold}}^{(i)}$ depends on the probability of $P_{\text{fold}}^{(j)}$. Fast iterative solvers for linear systems can be used to compute P_{fold} for all nodes. For their initial work, Apaydin *et al.* used the Jacobi method as their linear system solver, but noted that other approaches might provide faster performance.

SRS has been applied to the ColE1 repressor of primer and the homeodomain of Engrailed, a developmental protein, which are stored in the Protein Data Bank [45] as 1rop and 1hdd, respectively [19]. The vector model described in section 2.1 was used to represent the degrees of freedom. With this model, 1rop has 6 degrees of freedom and 1hdd has 12 degrees of freedom. Energy was computed by the H-P energy model [30] mentioned previously. P_{fold}, was computed for about 45 randomly selected conformations using SRS and using MC simulations. The correlation between the P_{fold} values of the two methods quickly converged to 1 as the number of nodes was increased, but SRS was roughly four orders of magnitude faster than the MC simulations. With SRS the roadmap captures a substantial sampling of all folding and unfolding pathways simultaneously, and P_{fold} was computed for *all* nodes, not just the 45 that were randomly selected. Thus, SRS appears to be a promising alternative to running many independent MC simulations for examining protein folding behavior.

In recent work, it has been demonstrated that SRS can be used to estimate the *transition state ensemble* and (TSE) and *folding rate* of proteins, as well as the Φ*-values* of residues [16]. All of these values are of interest because they are quantities that can be measured by laboratory experiment, and thus can be used to verify how well a simulation method such as SRS models the true behavior of a protein. Additionally, the TSE, if accurately determined, can provide insight into the overall folding mechanism of the protein.

The TSE is the set of conformations that represent the peak of the energy barrier that must be crossed by the protein in transitioning between the unfolded and native states. Alternatively, they are the states whose *true* P_{fold} is 0.5, the structures that have an equal probability of proceeding either to the folded or unfolded state. To account for modeling error, the TSE is taken to be the set of all conformations with P_{fold} between 0.45 and 0.55.

Apaydin *et al.* tested the method's ability to calculate the folding rate on a test set of 16 proteins, and compared the results with the dynamic programming algorithm of Garbuzinskiy *et al.* [46]. Intuitively, the folding rate is the fraction of unfolded molecules in some bulk set that transition to the folded state per unit of time. SRS-based estimates of the folding rate were found to correlate well with experimentally-determined values, and were consistently lower than those found by the other method. This indicates a consistent and significant difference between the transition state ensembles found by the two methods, and therefore, their predicted folding rates. The difference appeared to be due to a less restrictive definition of the TSE by the dynamic programming method. 80 percent of the structures identified as members of the TSE by the dynamic programming method were not considered part of the TSE by SRS. The more restricted set found by SRS led to more accurate estimation of measurable folding properties.

Φ-values are per-residue numbers between 0 and 1 indicating the degree to which the corresponding residue has reached its native conformation in the transition state of the protein [47]. They are measured in the laboratory by mutating specific residues of the protein and determining the effect of each mutation on its folding rate, and therefore, indirectly, the free energies of intermediate structures in the folding process. A Φ-value of 1 indicates that the mutation affects the folded state and transition state by the same amount, and that the transition state of that residue therefore is essentially the same as the folded state. A Φ-value of 0 means the residue is unfolded in the transition state.

The developers of SRS found Φ-values for each residue of their 16-protein test set [16]. The results were mixed, but promising. For some proteins, such as CheY and the RNA binding domain of U1A, their results correlated well with experiment, but their average error for Φ-values of the whole set of proteins was 0.21, which is quite large given the 0 to 1 range of Φ-values. Some of this error may be accounted for by the difference between the true free energy variation of folding, as measurable in a laboratory, versus the approximation of free energy used in simulations.

3.3 Markovian State Models

A different way to construct a roadmap is by sampling small MD/MC *trajectories* rather than individual conformations, generating a Markovian State Model (MSM) [15]. The use of MD/MC simulations for sampling suggests, among other things, that it is reasonable to expect that the resulting samples will have a realistic distribution of energies consistent with the predictions of statistical mechanics.

Suppose an initial MD or MC simulation trajectory starts in the folded state and ends in the unfolded state. Let $\{c_0, c_1, \ldots, c_n\}$ be a sequence of conformations along this trajectory separated by some fixed time step. A conformation c_i is selected uniformly at random from this sequence and a new MD/MC simulation is started from here. If the simulation does not reach the folded or unfolded state within some time limit, the trajectory is rejected. Otherwise, the trajectory is kept and a new current trajectory is created. Let the generated trajectory be denoted by $\{c'_0, c'_1, \ldots, c'_m\}$. If c'_m is in the folded state, the current trajectory becomes $\{c'_m, c'_{m-1}, \ldots, c'_0, c_i, \ldots, c_n\}$. If c'_m is in the unfolded state, the current trajectory becomes $\{c_1, c_2, \ldots, c_i, c'_0, c'_1 \ldots, c'_m\}$. Again, a conformation is selected uniformly at random from the current trajectory and this procedure of generating new trajectories is repeated a set number of times.

Each conformation and each transition in each sampled trajectory is represented by a node and an edge, respectively, in the roadmap. Each edge has associated with it a simulation time t_{ij} required to make the corresponding transition. The trajectories are simulated such that this timestep between adjacent conformations in the trajectory is constant. How this is done depends on the type of simulation being run. Each edge also has a probability P_{ij} that is initialized to 1. The next step is to merge nodes that are within some cut-off distance of each other, because they represent the same conformation. This step amounts to clustering of the nodes into conformational substates. To merge two nodes, one of the nodes is removed from the roadmap and all of its edges are added to the node it is merged with. If this results in multiple edges between a pair of nodes, the edges need to be merged as well. The probability and time of the merged edge are defined as:

$$P_{ij}^{\text{new}} = P_{ij}^1 + P_{ij}^2, \qquad t_{ij}^{\text{new}} = \frac{P_{ij}^1 t_{ij}^1 + P_{ij}^2 t_{ij}^2}{P_{ij}^1 + P_{ij}^2}.$$

After all nodes are merged that are within the cut-off distance of each other, the probabilities are renormalized so that the sum of the probabilities of all outgoing edges at a node is equal to 1. Singhal *et al.* [15] show that it is possible to derive a roadmap for a different temperature simply by reweighting the edges.

As with SRS, one can apply standard Markov chain techniques to compute P_{fold} from the roadmap. One can also compute the average time it takes to reach the folded state. The validity of this roadmap construction method was tested on a 2-dimensional artificial model system and on a small protein, the 12-residue tryptophan zipper beta hairpin, TZ2. TZ2 has previously been simulated on Folding@Home [48]. Some of this data was used to build a stochastic roadmap. The predicted P_{fold} values and the average times to reach the folded state were in agreement with experimental data.

One problem with both the SRS and the MSM method is that, because a roadmap of a conformation space is a discretization of a continuous space, the transition probabilities between nodes are only an approximation of reality. In a finite set of simulations, some states and transitions that occur with relatively low probability may never be sampled. Because the transition probabilities out of each node are forced to sum to 1, the transitions that are found are overrepresented due to the absence of others. This can lead to error in the computation of ensemble properties, including the predicted folding rate.

The developers of the MSM method proposed a method to estimate the error in the set of transition probabilities found by their sampling, and therefore the error (or uncertainty) in their calculated folding rates [14]. Furthermore, by isolating which states contribute the most to this uncertainty, it becomes possible to adaptively select which states to generate sample simulations from at each step in building the roadmap so as to minimize the final uncertainty of the folding rate.

In analysis of MSMs, the folding rate is measured by estimating the mean first passage time (MFPT) from the unfolded state, x_1, to the folded state. This requires estimation of the MFPT, x_i, for all nodes in the roadmap, as follows:

$$x_i = \begin{cases} \Delta t + \sum_{j=1}^{K} x_j p_{ij} & i \neq K, \\ 0 & i = K, \end{cases}$$

where K is the index of the folded state, Δt is the size of the time interval between successive structures in the simulations used to construct the MSM, and p_{ij} is the probability of transitioning from state i to state j in time Δt. The MFPT from the first state, x_1, can be used to estimate the folding rate of the protein under the simulated conditions.

The problem is that it is not possible to determine the exact values of p_{ij}, and therefore not possible to calculate exact values of MFPT. The maximum likelihood estimate, given the roadmap built through series of simulations, is $\hat{p}_{ij} = \frac{z_{ij}}{n_i}$, where z_{ij} is the observed number of transitions from state i to state j, and n_i is the total number of transitions out of state i. The observations z_{ij} follow a multinomial distribution that depends on the true transition probabilities. Ideally, the method would be able to estimate not just the most likely transition probabilities for a state, but also the distribution of all possible sets of transition probabilities, and therefore, our uncertainty of these estimates. Singhal *et al.* [14] show that this uncertainty follows a Dirichlet distribution, and based on that observation, provide a number of algorithms for finding the distribution of x_1, and therefore estimating the error of the calculated MFPT.

The basic idea of all of the algorithms is to sample a set of transition probabilities from a Dirichlet or approximation of a Dirichlet distribution whose parameters are based on the observed transition counts. Distributions for x_i, and, for MFPT, specifically x_1, are then inferred from the distributions of these samples. For details of the algorithms, please see the original paper [14].

The resulting uncertainty distribution for x_1 is a multivariate normal distribution, with calculable mean and variance. This distribution expresses how much confidence may be placed in the estimate of MFPT, but it also has implications for the construction of MSMs. It is possible to break the variance down into contributions from each state in the roadmap, and furthermore, to estimate the amount by which the variance due to any given state will decrease given some number of new MD/MC simulations starting from that state. The selection of which state to use for the next simulation need no longer be uniform at random, as described initially, but can instead be based on which choice of state is most likely to reduce the overall uncertainty of the MFPT by the greatest amount. This greatly increases the confidence of folding rate estimates and other properties calculated from an MSM generated by a set number of MD/MC simulations, versus undirected sampling.

Singhal *et al.* validated their error analysis method by again testing it on a set of simulations of TZ2, with a total of 87 distinct states. Using this example, they verified that all error estimation methods give comparable results for the mean and variance of the MFPT and that using the error estimates for adaptively focusing their sampling gave them a 20-fold improvement in certainty of their estimate of the MFPT for a given number of samples.

4 Discussion

Roadmap methods have been developed in recent years to study how a protein folds into its final known configuration. These roadmaps are generated by sampling conformations of a protein and connecting the sampled configurations in a number of ways. The variety of methods for generating and connecting roadmap nodes can only be expected to increase as time goes on. The same kind of growth was observed when roadmap methods became popular in robotics for solving the robot motion planning problem as researchers began to understand how to better target their methods to the characteristics of the problems being address (see [31]). All existing approaches struggle to understand how to use energy estimates in the construction of the roadmap and the interpretation of the results. A number of questions is raised about how to compute the free energy for proteins of interest, which is a serious issue and a topic in need of further study.

Although the performance of roadmap methods is often compared to MD/MC methods, for now roadmaps are not necessarily meant to be a substitute for MD/MC simulations. Rather, the hope is that with a simplified energy model and clever sampling techniques roadmap methods could quickly provide a coarse view of the energy landscape. Of course, much depends on the energy function used. The areas of interest identified in this landscape can provide a starting point for traditional MD/MC simulations.

Roadmap methods have also been applied to the study of other biological problems, including docking. In docking, the goal is to find low-energy conformations of a receptor-ligand complex. Recent examples of this work include [36, 37]. Structure prediction is another area where roadmap methods have been applied [38, 39]. By a combination of cleverly sampling and pruning conformations Brunette and Brock [38, 39] build up a compact model of the molecular energy landscape for a given protein. Finally, a roadmap-based method for the generation of loop conformations was developed in [40]. Clearly, there are attractive features in a roadmap-based approach for exploring high-dimensional spaces arising from geometric problems which has prompted researchers to use them in a variety of biological problems. Although roadmap-based methods are well understood in robotic problems, it is the authors' belief that a number of issues that mainly relate to the interplay of energy and geometry are still poorly understood for biological problems. Nevertheless, promising results are emerging that will no doubt fuel further advancements.

References

[1] M. Gruebele. Protein folding: the free energy surface. *Current Opinion in Structural Biology*, 12:161–168, 2002.

[2] T. Head-Gordon and S. Brown. Minimalist models for protein folding and design. *Current Opinion in Structural Biology*, 13:160–167, 2003.

[3] X. Zhuang and M. Rief. Single-molecule folding. *Current Opinion in Structural Biology*, 13:88–97, 2003.

[4] M. Vendruscolo and E. Paci. Protein folding: bringing theory and experiment closer together. *Current Opinion in Structural Biology*, 13:82–87, 2003.

[5] C. M. Dobson. Protein folding and misfolding. *Nature*, 426:884–890, 2003.

[6] J. N. Onuchic and P. G. Wolynes. Theory of protein folding. *Current Opinion in Structural Biology*, 14:70–75, 2004.

[7] C. M. Dobson. Principles of protein folding, misfolding and aggregation. *Seminars in Cell and Developmental Biology*, 15:3–16, 2004.

[8] M. S. Apaydin. *Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion.* PhD thesis, Stanford University, Stanford, CA 94305 USA, Aug 2004.

[9] S. L. Thomas, X. Tang, L. Tapia, and N. M. Amato. Simulating protein motions with rigidity analysis. In *Proc. of the ACM Intl. Conf on Research in Computational Molecular Biology (RECOMB)*, pages 394–409, 2006.

[10] S. Thomas, G. Song, and N. M. Amato. Protein folding by motion planning. *Physical Biology*, 2:S148–S155, 2005.

[11] 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. *J. Comp. Bio.*, 10(3–4):239–255, 2003.

[12] G. Song. *A Motion Planning Approach to Protein Folding.* PhD thesis, Dept. of Computer Science, Texas A&M University, December 2003.

[13] N. M. Amato and G. Song. Using motion planning to study protein folding pathways. *J. Comp. Bio.*, 9(2):149–168, 2002.

[14] N. Singhal and V. S. Pande. Error analysis and efficient sampling in Markovian state models for molecular dynamics. *The Journal of Chemical Physics*, 123(20):204909, 2005.

[15] N. Singhal, C. D. Snow, and V. S. Pande. Using path sampling to build better Markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *J. Chem. Physics*, 121(1):415–425, July 2004.

[16] T.-H. Chiang, M. S. Apaydin, D. L. Brutlag, D. Hsu, and J.-C. Latombe. Predicting experimental quantities in protein folding kinetics using stochastic roadmap simulation. *Proc. of the ACM Intl. Conf on Research in Computational Molecular Biology (RECOMB)*, pages 410–424, 2006.

[17] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, and J.-C. Latombe. Stochastic conformational roadmaps for computing ensemble properties of molecular motion. In J. D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, editors, *Algorithmic Foundations of Robotics V*, pages 131–147. Springer, 2004.

[18] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe, and C. Varma. Stochastic roadmap simulation: an efficient representation and algorithm for analyzing molecular motion. *J. Comp. Bio.*, 10(3–4):257–281, 2003.

[19] M. S. Apaydin, C. E. Guestrin, C. Varma, D. L. Brutlag, and J.-C. Latombe. Stochastic roadmap simulation for the study of ligand-protein interactions. *Bioinformatics*, 18 Suppl 2:18–26, 2002.

[20] M. Karplus and J. Kuriyan. Molecular dynamics and protein function. *Proceedings of the National Academy of Science*, 102:6679–6685, 2005.

[21] D. R. Ripoll, J. A. Vila, and H. A. Scheraga. Folding of the villin headpiece subdomain from random structures. Analysis of the charge distribution as a function of pH. *Journal of Molecular Biology*, 339(4):915–925, 2004.

[22] W. F. van Gunsteren and H. J. C. Berendsen. Computer simulation of molecular dynamics: methodology, applications and perspectives in chemistry. *Angewandte Chemie International*, 29(9):992–1023, 1990.

[23] T. Huber, A. E. Torda, and W. F. van Gunsteren. Local elevation: a method for improving the searching properties of molecular dynamics simulation. *Journal of Computer Aided Molecular Design*, 8(6):695–708, 1994.

[24] B. G. Schulze, H. Grubmueller, and J. D. Evanseck. Functional significance of hierarchical tiers in carbonmonoxy myoglobin: conformational substates and transitions studied by conformational flooding simulations. *Journal of the American Chemical Society*, 122(36):8700–8711, 2000.

[25] Y. Zhang, D. Kihara, and J. Skolnick. Local energy landscape flattening: parallel hyperbolic Monte Carlo sampling of protein folding. *Proteins: Structure, Function, and Bioinformatics*, 48(2):192–201, 2002.

[26] K. Lindorff-Larsen, R. B. Best, M. A. DePristo, C. M. Dobson, and M. Vendruscolo. Simultaneous determination of protein structure and dynamics. *Nature*, 433(7022):128–132, 2005.

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

[28] A. D. MacKerell, Jr. Empirical force fields for biological macromolecules: Overview and issues. *J. Comp. Chemistry*, 25(13):1584–1604, October 2004.

[29] M. Zhang and L. E. Kavraki. A new method for fast and accurate computation of molecular conformations. *Journal of Chemical Information and Computer Sciences*, 42:64–70, 2002.

[30] 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(8):769–778, Aug 1995.

[31] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, 2005.

[32] J.-C. Latombe. *Robot Motion Planning*, chapter 7, pages 295–353. Kluwer, Dordrecht; Boston, 1991.

[33] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *Intl. J. of Robotics Research*, 20(5):378–400, May 2001.

[34] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Intl. J. of Computational Geometry and Applications*, 9(4-5):495–512, 1999.

[35] A. Ladd and L. E. Kavraki. Fast exploration for robots with dynamics. In *Workshop on the Algorithmic Foundations of Robotics*, 2004.

[36] M. Moll, M. D. Schwarz, A. Heath, and L. E. Kavraki. On flexible docking using expansive search. Technical Report 04-443, Rice University, Houston, TX, 2004.

[37] J. Cortés, T. Siméon, V. R. de Angulo, D. Guieysse, M. Remauld-Siméon, and V. Tran. A path planning approach for computing large-amplitude motions of flexible molecules. *Bioinformatics*, 21 Suppl. 1:i116–i125, 2005.

[38] T. J. Brunette and O. Brock. Improving protein structure prediction with model-based search. *Bioinformatics*, 21 Suppl. 1:i66–i74, 2005.

[39] T. J. Brunette and O. Brock. Model-based search to determine minima in molecular energy landscapes. Technical Report 04-48, Dept. of Computer Science, University of Massachusetts, Amherst, MA, 2005.

[40] J. Cortés, T. Siméon, M. Remaud-Siméon, and V. Tran. Geometric algorithms for the conformational analysis of long protein loops. *J. Comp. Chemistry*, 25(7):956–967, May 2004.

[41] A. P. Singh, J.-C. Latombe, and D. L. Brutlag. A motion planning approach to flexible ligand binding. In *Proc. 7th ISMB*, pages 252–261, 1999.

[42] G. N. Ramachandran and V. Sasisekharan. Conformation of polypeptides and proteins. *Adv. Protein. Chem*, 23:283–438, 1968.

[43] T. H. Cormen, C. E. Leiserson, R. R. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, second edition, 1990.

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

[45] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[46] S. O. Garbuzynskiy, A. V. Finkelstein, and O. V. Galzitskaya. Outlining folding nuclei in globular proteins. *Journal of Molecular Biology*, 336:509–525, 2004.

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

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