Adaptive Local Learning in Sampling Based Motion Planning for Protein Folding

Chinwe Ekenna, Shawna Thomas, Nancy M. Amato

***Abstract*—Motivation: Simulating protein folding motions is an important problem in computational biology. Motion planning algorithms such as Probabilistic Roadmap Methods (PRMs) have been successful in modeling the protein folding landscape. PRMs and variants contain several phases (i.e., sampling, connection, and path extraction). Global machine learning has been applied to the connection phase but is inefficient in situations with varying topology, such as those typical of folding landscapes.**

Results: We present a *local* learning algorithm that considers the past performance near the current connection attempt as a basis for learning. It is sensitive not only to different types of landscapes but also to *differing regions* in the landscape itself, removing the need to explicitly partition the landscape. We perform experiments on 23 proteins of varying secondary structure makeup with 52–114 residues. Our method models the landscape with better quality and comparable time to the best performing individual method and to global learning.

I. Introduction

Modeling the protein folding process is crucial in understanding not only how proteins fold and function, but also how they misfold triggering many devastating diseases (e.g., Mad Cow and Alzheimer's [7]). Since the process is difficult to experimentally observe, computational methods are critical.

Traditional computational approaches for generating folding trajectories such as molecular dynamics [21], Monte Carlo methods [9], and simulated annealing [20] provide a single, detailed, high-quality folding pathway at a large computational expense. As such, they cannot be practically used to study global properties of the folding landscape or to produce multiple folding pathways. Statistical mechanical models have been applied to compute statistics related to the folding landscape [28, 6]. While computationally more efficient, they do not produce individual pathway trajectories and are limited to studying global averages of the folding landscape.

Robotics-based motion planning techniques, including the Probabilistic Roadmap Method (PRM), have been successfully applied to protein folding [2, 3, 8]. They construct a roadmap, or model, of the folding landscape by sampling conformations and connecting neighboring ones together with feasible transitions using a simple local planner. They can generate multiple folding pathways efficiently (e.g., a few hours on a desktop PC) enabling the study of both individual folding trajectories and global landscape properties.

While promising, making good choices for each of the algorithmic steps remains difficult. Machine learning approaches have been used to dynamically decide which approaches to take for generating samples and connecting them together. These approaches generally learn *globally* and can perform well in homogeneous spaces or partitioned spaces where each partition is homogeneous. Preliminary work applied connection learning to protein folding simulations [12], but with no way to ensure a good partitioning of the landscape, the results were only comparable to methods with no learning involved.

We present *Local Adaptive Neighbor Connection* (ANC-local) that localizes learning to within the vicinity of the current conformation being connected. When choosing a connection method (i.e., the neighbor selection method and local planner combination), we first dynamically determine a neighborhood around the conformation under consideration. Then, the performance history within this neighborhood is used to bias learning. Our method adapts *both* over time and to local regions without any prior knowledge about the methods involved. This approach has been successfully used in robotics [13], and here we adapt it to protein folding.

We compare ANC-local's performance to three distance-based connection methods and to global learning over 23 proteins of varying secondary structure makeup with 52–114 residues. We examine both the time to build roadmaps and the resulting trajectory quality. No individual method is the best choice for all inputs. ANC-local generates better quality trajectories in comparable time than the best connection method for each individual input and outperforms global learning.

II. Preliminaries & Related Work

A. Protein Model

Proteins are sequences of amino acids, or residues. We model the protein as a linkage where only the ϕ and ψ torsional angles are flexible, a standard modeling assumption [26]. A potential energy function models the many interactions that affect the protein's behavior [21]. This function helps quantify how energetically feasible a given conformation is. In this work, we employ a coarse-grained potential function [2]. If the atoms are too close to each other (less than 2.4Å in sampling and 1.0Å in connecting), the conformation is unfeasible; otherwise, the energy is calculated by:

$$U_{tot} = \sum_{constraints} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_c\} + E_{hp} \quad (1)$$

This research supported in part by NSF awards CNS-0551685, CCF-0833199, CCF-1423111, CCF-0830753, IIS-0916053, IIS-0917266, EFRI-1240483, RI-1217991, by NIH NCI R25 CA090301-11, and by the Schlumberger Faculty for the Future Fellowship. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

C. Ekenna, S. Thomas, and N. M. Amato are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, 77843, USA. {cekenna, sthomas, amato}@cse.tamu.edu

where K_d is 100 kJ/mol, d_i is the length on the ith constraint, E_{hp} is the hydrophobic interaction, and $d_0 = d_c = 2$Å as in [21]. The coarse grain model has been shown to produce qualitatively similar results as all-atoms models faster [32].

B. PRM for Protein Folding

The Probabilistic Roadmap Method (PRM) [18] is a robotics motion planning algorithm that first randomly samples robot (or protein) conformations, retains valid ones, and then connects neighboring samples together with feasible motions (or transitions). To apply PRMs to proteins, the robot is replaced with a protein model and collision detection computations are replaced with potential energy calculations [2, 3, 8, 1].

1) Sampling: Protein conformations, or samples, are randomly generated with bias around the native state, the functional and most energetically stable state. Samples are iteratively perturbed, starting from the native state, and retained if energetically feasible by the following probability:

$$P(q) = \begin{cases} 1 & \text{if } E(q) < E_{min} \\ \dfrac{E_{max} - E(q)}{E_{max} - E_{min}} & \text{if } E_{min} < E(q) \leq E_{max} \\ 0 & \text{if } E(q) > E_{max} \end{cases} \quad (2)$$

where E_{min} is the energy of the open chain and E_{max} is $2E_{min}$. We use rigidity analysis to focus perturbations on flexible portions as detailed in [34].

2) Connection: Once a set of samples is created, they must be connected together with feasible transitions to form a roadmap, or model of the folding landscape. Connecting all possible pairs of samples is computationally unfeasible, and it has been shown that only connecting the k-closest neighbors results in a roadmap of comparable quality [27].

Given a pair of samples, we compute a transition between them by a straight-line interpolation of all the ϕ and ψ torsional angles. Straight-line local planning involves the fewest number of intermediates to check for validity and has been shown to be a sufficient measure of transition probability; i.e., it can accurately predict secondary structure formation order [2, 32]. We assign an edge weight to reflect the energetic feasibility of the transition as $\sum_{i=0}^{n-1} -log(P_i)$ where P_i is the probability to transit from intermediate conformation c_i to c_{i+1} based on their energy difference $\Delta E_i = E(c_{i+1}) - E(c_i)$:

$$P_i = \begin{cases} e^{\frac{-\Delta E_i}{kT}} & \text{if } \Delta E_i > 0 \\ 1 & \text{if } \Delta E_i \leq 0 \end{cases} \quad (3)$$

where k is the Boltzmann constant and T is the temperature. This allows the most energetically feasible paths to be extracted by standard shortest path algorithms.

3) Validation by Secondary Structure Formation Order: Proteins are composed of secondary structure elements (i.e., α-helices and β-strands). Experimental methods, such as hydrogen exchange mass spectrometry and pulse labeling, can investigate protein folding by identifying which parts of the structure are most exposed or most protected [38]. From this data, one can infer the secondary structure formation order.

In [2, 26, 32] we compared the secondary structure formation order of folding pathways extracted from our maps to experimental results [22] by clustering paths together if they have the same formation ordering. We return a stable roadmap when the distribution of secondary structure formation orderings along the folding pathways in the graph stabilizes, i.e., the percentage of pathways following a given ordering does not vary between successive graphs by more than 30%. As our roadmaps contain multiple pathways, we estimate the probability of a particular secondary structure formation order occurring by the percentage of roadmap pathways that contain that particular formation order. The roadmap corroborates experimental data when the dominant formation order (i.e., the one with the greatest percentage) is in agreement.

C. Candidate Neighbor Selection Methods

Recall that only neighboring (or nearby) samples are attempted for connection because it is unfeasible to attempt all possible connections. Typically, conformations that are more similar are more energetically feasible to connect.

There have been a number of methods proposed for locating candidate neighbors for connection. The most common is the k-closest method which returns the k closest neighbors to a sample using a distance metric. This can be implemented in a brute force manner taking $O(k \log n)$-time per node, totaling $O(nk \log n)$-time for connection. A similar approach is the r-closest method which returns all neighbors within a radius r of the node as determined by some distance metric.

Other methods use data structures to more efficiently compute nearest neighbors. *Metric Trees* [35] organize the nodes in a spatial hierarchical manner by iteratively dividing the set into two equal subsets resulting in a tree with $O(\log n)$ depth. However, as the dataset dimensionality increases, their performance decreases [23]. *KD-trees* [4] extend the intuitive binary tree into a D-dimensional data structure which provides a good model for problems with high dimensionality. However, a separate data structure needs to be stored and updated.

Approximate neighbor finding methods address the running time issue by instead returning a set of approximate k-closest neighbors. These include spill trees [23], MPNN [39], and Distance-based Projection onto Euclidean Space [30]. These methods usually provide a bound on the approximation error.

In this paper, we work with proteins with a higher dimensionality than approximate methods can handle. These proteins range from 52-114 residues (104 to 228 degrees of freedom), and it would be impractical to employ approximate methods to them. Note, however, that there is nothing inherent in our approach that precludes the use of approximate methods.

D. Distance Metrics

The distance metric plays an important role in determining the best connections to attempt. It is a function δ that computes some "distance" between two conformations $a = \langle a_1, a_2, \ldots, a_d \rangle$ and $b = \langle b_1, b_2, \ldots, b_d \rangle$, i.e., $\delta(a, b) \rightarrow \mathbb{R}$, where d is the dimension of a conformations. A good distance metric generally predicts how likely it is that a pair of nodes can be successfully connected. Their success is dependent on the nature of the problem studied. We use the following set of distance metrics commonly used for motion planning:

Euclidean Distance Metric. The Euclidean distance metric captures the amount of physical movement (around the torsional angles) that conformation a would undertake to move to a conformation b. This distance is computed by measuring the difference in the ϕ and ψ angle pairs of the two conformations:

$$\delta_{\text{Eucl}}(a,b) = \sqrt{\frac{(\phi_1^a-\phi_1^b)^2+(\psi_1^a-\psi_1^b)^2+\ldots+(\phi_n^a-\phi_n^b)^2+(\psi_n^a-\psi_n^b)^2}{2n}}. \quad (4)$$

Cluster Rigidity Distance Metric. Rigidity analysis [17] computes which parts of a structure are rigid and flexible based on the constraints present. It may be used to define a rigidity map r, which marks residue pairs i, j if they are in the same rigid cluster. Rigidity maps provide a convenient way to define a rigidity distance metric, between two conformations a and b where n is the number of residues:

$$\delta_{\text{Rig}}(a,b) = \sum_{0\le i<j\le 2n} (r_a(i,j) \neq r_b(i,j)). \quad (5)$$

More details may be found in [34].

Root Mean Square Distance Metric. The protein model has 6 atoms for each amino acid. Thus, a protein with n amino acids will have $6n$ atoms. Denoting the coordinates of these atoms as x_1 to x_{6n}, the root mean square distance (RMSD) between conformations a and b is:

$$\delta_{\text{RMSD}}(a,b) = \sqrt{\frac{(x_1^a - x_1^b)^2 + (x_2^a - x_2^b)^2 + \ldots + (x_{6n}^a - x_{6n}^b)^2}{6n}}. \quad (6)$$

Least RMSD (lRMSD) is the minimum RMSD over all rigid body superpositions of a and b.

E. Adaptive Neighbor Connection (ANC-global)

Prior work [11] adaptively selects the appropriate connection method to use over time. As the roadmap is built, it records the performance of several connection methods and with this history, decides which to employ by maintaining a selection probability for each.

The main weakness is that it bases its decisions on the performance of connection methods over the *entire* landscape. This is problematic in protein landscapes that are naturally heterogeneous. Therefore, to obtain better results, it became necessary to first partition the space into smaller (and hopefully homogeneous) regions [11]. This puts greater burden on the user, particularly as the dimensionality of the problem increases. While ANC-global was applied to proteins, its performance was limited and so a *local* learning approach is needed.

III. The Local Learning Approach

We apply *Local Adaptive Neighbor Connection* (ANC-local) to protein folding. It localizes learning to within the vicinity of the current conformation being connected. When choosing a connection method, the current conformation's neighborhood is dynamically determined. This neighborhood is defined as the set of nearest neighbors given by some distance metric.

We use the performance history of only those connection attempts within this neighborhood to bias learning. Thus, our method adapts both spatially and temporarily, and no prior knowledge about the connection method involved is needed. This approach has been introduced for robotic motion planning [13], and here we adapt it to simulate the folding process.

For proteins, we measure performance as a function of the edge weights in the roadmap and the time needed to construct a stable roadmap. We want to balance both compute time and trajectory quality where quality may be inferred from the edge weights (i.e., their energetic feasiblity). Performance is measured only from the dynamically determined neighborhood so learning is continuous and localized.

A. Example

Figure 1 shows an example energy landscape and roadmap. The roadmap is constructed with two candidate connection methods: CM_A (yellow/light) and CM_B (blue/dark). Overall, the most successful connection method is CM_A (with more yellow/light edges). However, in the left region of the landscape, CM_B is much more successful. When connecting node q (in green) to the roadmap, it is important to take locality into account. A global learning method, such as ANC-global, would select CM_A to connect q, but this would be a poor choice. A local learning method, such as ANC-local, would instead choose CM_B to connect q because CM_B is more successful there.

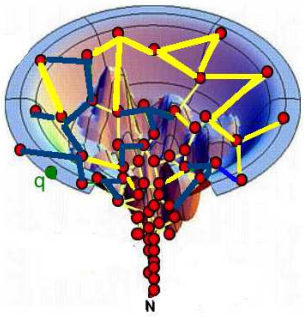


Fig. 1. Two connection methods are used to build a roadmap on the protein's energy landscape: CM_A (yellow/light) and CM_B (blue/dark).

B. Method

Algorithm 1 describes the ANC-local algorithm as introduced in [13]. We initialize all the methods M to the uniform probability and determine the local learning region as defined by the set of nearest neighbors using NF_{local} in D, where D is a tuple containing the connection method, reward, and cost. For each determined neighbor, we update the probability using the UpdateProbability function in Algorithm 2 and make a connection based on the chosen connection method cm.

Algorithm 1 ANC-local(D, M, NF_{local})

1: Let P_q be a set of probabilities initialized to the uniform distribution, D be data containing tuples $(m, reward, cost)$, NF_{local} be a neighbor finding method, and M be a set of connection methods such that $|P_q|$ = $|M|$ and $cm \in M$.
2: Let L be the learning region defined as the set of nearest neighbors to q given by NF_{local} in D.
3: **for** each $n \in L$ **do**
4: P_q = UpdateProbability($n.cm$, $n.reward$, $n.cost$)
5: **end for**
6: Select cm based on P_q.
7: Make connection using cm.

Algorithm 2 UpdateProbability(cm, $reward$, $cost$)

1: $reward \leftarrow$ Update $reward$ using Eqs. 8 and 9
2: $w \leftarrow$ Update weight using $reward$ and cm in Eq. 10
3: $P^* \leftarrow$ Calculate without $cost$ using w in Eq. 7
4: $P \leftarrow$ Calculate using P^*, cm and cost in Eq. 11
5: **return** P

The UpdateProbabilty function (Algorithm 2) is used to continually calculate and update the probabilities of the connection methods. This is where performance is monitored and reinforcement learning takes place.

Potential energy computations take up a large portion of the total computation time and thus are a good measure of cost. Here, we calculate the cost as the number of potential energy calls incurred by the connection method.

ANC-local maintains a weight for each connection method similar to Hybrid PRM [15] but reconstructed to handle potential energy calculations. These weights keep track of the past performance of each connection method. ANC-local initializes each weight w_i to 1. Based on the weights, ANC-local computes in a step-wise manner a probability p_i^* for cm_i without considering the cost:

$$p_i^* = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{m} w_j(t)} + \gamma\frac{1}{m}, i = 1, 2, ..., m, \quad (7)$$

where $w_i(t)$ is the weight of cm_i in step t, t is the current connection attempts made, and γ is a fixed constant. The probability p_i^* is a weighted sum of the relative weight of cm_i and the uniform distribution. This ensures that each connection method has some chance of being selected.

Let x_i be the reward for the cm_i that was selected:

$$x_i = \alpha + (1-\alpha)(1 - \frac{y_i(t) - miny_i(t)}{maxy_i(t) - miny_i(t)}) \quad (8)$$

where $y_i(t)$ = current edge weight, $miny_i(t)$ = minimum edge weight recorded during the current step, $maxy_i(t)$ = maximum edge weight recorded during the current step, and α = a constant value used to normalize the reward. All other rewards for that time step are 0. The reward is thus a function of the edge quality (weight) and the local planner's success.

To update the weights, we first take into account an adjusted reward that is not dependent on the cost accrued:

$$x_i^* = x_i / p_i^*, i = 1, 2, ...m. \quad (9)$$

Then we update the weights for all the connection methods:

$$w_i(t+1) = w_i(t) \exp \frac{\gamma x_i^*}{m}, i = 1, 2, ...m. \quad (10)$$

The new weight is the current weight multiplied by a factor that depends on the reward received. The exponential factors enable the weights to adapt quickly.

We now include the cost in the selection probability:

$$p_i = \frac{\frac{p_i^*}{c_i}}{\sum_{j=1}^{m} \frac{p_i^*}{c_j}}, i = 1, 2, ...K. \quad (11)$$

where c_i is the average cost of attempting to connect i.

IV. Experimental Results

In this section, we investigate the performance of ANC-local (local learning), ANC-global (global learning), and individual connection methods to model the folding landscape of 23 proteins. Individual connection methods are k-closest neighbor selection using either Cluster, Euclidean, or lRMSD distance metric (see Section II-D). ANC-global and ANC-local use these methods as their learning set.

We first establish each method's ability (individual connection methods, global learning, and local learning) to validate against experimental data when available. We next examine the quality of the resulting folding pathways and the time required by each individual method. We also look at the cummulative performance of these metrics and show how ANC-local's learning decisions corroborate with the individual connection method performance outside of the learning framework. In addition, we compare ANC-local's local learning performance against ANC-global's global learning approach.

A. Experimental Setup

We study 23 proteins (see Table I) with 52–114 residues. This set contains α, β, and mixed proteins that were also studied by [10] and have experimentally determined secondary structure formation orders [24]. The protein structures were obtained from the Protein Data Bank [5].

For all experiments, we generate conformations using iterative sampling based on rigidity analysis [34]. For all connection methods, we use a straight line local planner and attempt to connect to the 20 nearest neighbors. For ANC-local, we set NF_{local} to be the 40 nearest neighbors based on Euclidean distance. This resulted in the best performance in preliminary experiments. We stop construction once we have a stable roadmap (see Section II-B3).

Metrics are computed as follows:

- *Secondary Structure Formation Order:* We compare, when available, the secondary structure formation order predicted by each method to experimental data. See Section II-A for details.

Protein Name	PDB ID	Length	Secondary Structure
Rubredoxin	1RDV	52	$2\alpha + 3\beta$
Ferredoxin	1FCA	55	$2\alpha + 4\beta$
Protein G	1PGA	56	$1\alpha + 4\beta$
Protein G Variant	NUG1	57	$1\alpha + 3\beta$
Protein G Variant	NUG2	57	$1\alpha + 3\beta$
Alpha-Spectrin SH3 Domain	1SHG	57	$1\alpha + 5\beta$
Human FYN	1NYF	58	$4\alpha + 1\beta$
Immunoglobulin G Binding Protein A	2SPZ	58	3α
Cardiotoxin III	2CRS	60	5β
Tick Antocoagulant peptide	1TCP	60	$2\alpha + 2\beta$
ADR1	2ADR	60	6β
Repressor Protein C1	1R69	63	4α
Chymotrypsin Inhibitor 2 variant	1COA	64	$1\alpha + 5\beta$
Chymotrypsin Inhibitor 2 variant	2CI2	65	$2\alpha + 5\beta$
Probable enterotoxin	2KRS	70	$1\alpha + 5\beta$
Regulatory Protein CRO	2CRO	71	5α
Protein L	2PTL	78	$1\alpha + 4\beta$
Procarboxy peptidase B	1PBA	81	$4\alpha + 3\beta$
Procarboxy peptidase A2	106X	81	$2\alpha + 3\beta$
ACYL-CO Enzyme	2ABD	86	5α
Barnase	1YVS	106	$3\alpha + 5\beta$
Binase	1BUJ	109	$3\alpha + 6\beta$
DNA B Helicase	1JWE	114	6α

TABLE I
PROTEINS STUDIED.

PDB Identifier	Experimental Data	ANC-local	ANC-global	Cluster	Euclidean	lRMSD
1RDV	unavailable	same ordering				
1FCA	unavailable	same ordering				
1PGA	[19]	Y	Y	Y	Y	Y
NUG1	[29]	Y	Y	Y	Y	Y
NUG2	[29]	Y	Y	Y	Y	Y
1SHG	[36, 24]	Y	Y	Y	Y	Y
1NYF	[14, 31]	Y	Y	Y	Y	Y
2SPZ	unavailable	**different orderings**				
2CRS	[22]	Y	Y	Y	Y	Y
1TCP	unavailable	same ordering				
2ADR	unavailable	same ordering				
1R69	unavailable	same ordering				
1COA	unavailable	same ordering				
2CI2	[16]	Y	Y	Y	Y	Y
2KRS	[24]	Y	Y	Y	Y	Y
2CRO	unavailable	same ordering				
2PTL	[40]	Y	Y	Y	Y	Y
1PBA	unavailable	same ordering				
106X	[37]	Y	Y	Y	Y	Y
2ABD	[33]	Y	Y	**N**	**N**	**N**
1YVS	[25]	Y	Y	Y	Y	Y
1BUJ	unavailable	**different orderings**				
1JWE	unavailable	same ordering				
# Agree with Exp. / # Available		12/12	12/12	11/12	11/12	11/12

TABLE II
VALIDATION OF SECONDARY STRUCTURE FORMATION ORDER TO EXPERIMENTAL DATA WHEN AVAILABLE. PROTEINS ARE ORDERED BY PROTEIN LENGTH AS IN TABLE I.

- *Pathway Quality:* We define folding pathway quality as the weight of each edge (i.e., its energetic feasibility) multiplied by the dominance of that edge (i.e., the number of folding pathways that traverse it). This metric is important because it identifies how many edges with low energies are present and how frequently they are used.

B. Results and Discussion

We first establish each method's ability (or not) to produce folding trajectories that agree with available experimental data. We then compare the quality of each method's roadmaps and the time to construct them.

1) Validation by Secondary Structure Formation Order: Table II summarizes the comparison of each method's dominant secondary structure formation order. (Entries are ordered as appears in Table I by protein length.) Only the learning methods (ANC-global and ANC-local) produced the same dominant formation order as experiment for all proteins with available data. Individual methods were unable to reproduce the ordering from experimental data for 2ABD. Thus, in some cases learning is required for correctness.

When experimental data was not available, all methods produced the same ordering for 9 proteins and different orderings for 2 proteins (2SPZ and 1BUJ). Upon examination of the 2 proteins that methods disagree on, we find that ANC-local, ANC-global, and Cluster are always in agreement and Euclidean and lRMSD are always in agreement. Additionally, disagreements only occur at the end of the pathway; all methods agree on the order of the first elements to form. Specifically, all methods find that the central α-helix forms first in 2SPZ and disagree on the relative ordering of the two terminal α-helices. Similarly, all methods find that β-strands 6, 5, 4, 3, 2 form first (and in that order) and disagree on the relative ordering of the three α-helices and the remaining β-strand for 1BUJ.

2) Quality vs. Time: Figure 2(a) shows the resulting folding pathway quality of each connection method, ANC-global, and ANC-local. Entries are ordered by ANC-local performance (and not by protein length). Recall that the aim is to generate pathways with low weight/energy. Only looking at individual connection method performance, we first see that no single connection method performs the best across all proteins: Cluster is the best choice for 7 proteins, Euclidean for 11 proteins, and lRMSD for 5 proteins. In addition, there is no correlation between individual connection method performance and secondary structure makeup or size. Thus, there is a clear need for learning.

It is not surprising then that learning methods outperform the best individual connection methods much of the time: ANC-global (pink bars) produces lower weighted pathways than Cluster, Euclidean, and lRMSD for 11 of the 23 proteins, and ANC-local (blue bars) for 19 of the 23 proteins. Notice, however, that the type of learning is important. ANC-local with its local learning is much more successful than ANC-global with its global approach. ANC-global outperforms ANC-local for only 1 protein in the set (2ADR) and even then the performance is only marginally better while ANC-local outperforms ANC-global by a large margin for many of the proteins. In fact, ANC-local is the best approach for 18 out of the 23 proteins studied. Note that the best performing method in the other 5 proteins is not the same (many of them are at the far right of Figure 2(a)): lRMSD produces lower weight pathways for 3 proteins (2KRS, 2ABD, and 1JWE), Euclidean for 1 (2CRO), and ANC-global for 1 (2ADR).

Additionally, in 17 of the 18 proteins where ANC-local produces the best quality, it produces significantly better quality than the other methods for 12 of the 18. We see an improvement of ANC-local over ANC-global in terms of quality for 20 of the 23 proteins studied. Of the 3 remaining

proteins (2ADR, 2CRO, 2KRS) where ANC-global performs better, ANC-local performance is comparable.

Figure 2(b) provides the time needed to build stable roadmaps for each method, ordered by protein length. ANC-local is the fastest for 6 of the proteins and the second fastest for 6, with 3 of those incurring less than 10% overhead. Thus, ANC-local performs as well as or better than the best performing method for 12 out of 23 proteins (52% of the time), while ANC-global performs best for only 3. Just as with quality, the best performing individual connection method varies between proteins: Euclidean is fastest for 11 proteins, Cluster for 2, and lRMSD for 1. Euclidean is most often the fastest method but is the best method in terms of quality for only 1 protein.

Finally, we look at each method's the cumulative performance. Figure 3 shows the ordered ranking of each connection method, ANC-global, and ANC-local across all 23 proteins. For each protein, we assign a rank from 1 to 5 (with 5 being the best) to each method for quality and time. The cumulative performance for each method is the average of these rankings.

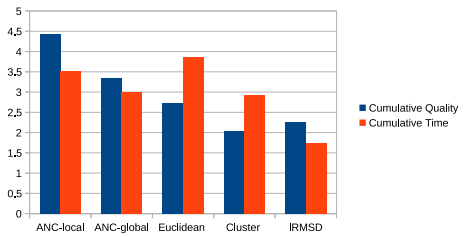


Fig. 3. The cumulative performance of each method as the average ranking from 1 to 5 (with 5 being the best) over all proteins studied. Entries are ordered by cumulative quality ranking.

ANC-local performs better than the other connection methods across the entire protein set in terms of quality and second best in terms of time. lRMSD, as expected, is the slowest. While ANC-local is not the fastest overall (Euclidean is), it does produce the best quality. ANC-local is the only method that is able to adapt locally to varying energy landscapes and thus yields higher quality roadmaps. ANC-global is the second best in terms of quality but third in terms of time. ANC-local outperforms ANC-global.

3) Inspection of ANC-local Learning Choices: Figure 4 shows the percentage at which ANC-local used each individual connection method in constructing stable roadmaps for each protein. Entries are ordered by Euclidean usage as it is most often selected across the entire set.

For many proteins, ANC-local favored a single connection method, but for some (1O6X, 1TCP – NUG1), it favored 2 connection methods, and for 2 proteins (2PTL and 1R69), it selected equally among all connection methods. When it did favor a subset of the connection methods, it selected the best individual method in both time and quality for 9 proteins, the best individual method in time only for 4 proteins, and the best individual method in quality only for 3 proteins.

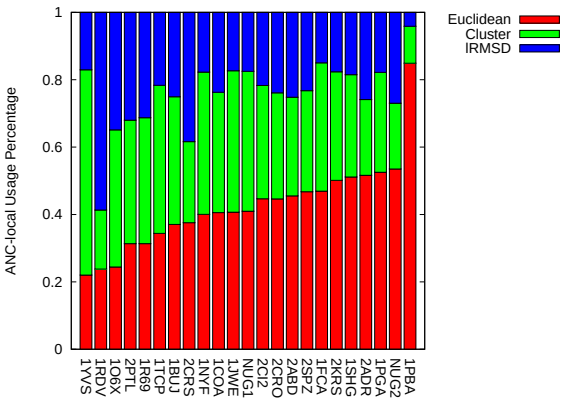


Fig. 4. The usage percentage for each connection method in ANC-local across all proteins studied. Entries ordered by Euclidean usage.

V. Conclusion

In this work, we present ANC-local, an algorithm that uses *local* learning to select appropriate connection methods in the context of PRM roadmap construction for protein folding. Our method monitors the performance and cost of various methods within the local neighborhood of the connecting conformation and adjusts their selection probabilities accordingly.

We have demonstrated a clear need for learning (i.e., ANC-global and ANC-local were the only methods to validate against all available experimental data) and showed that local learning is superior to global learning (i.e., ANC-local outperformed all other methods in terms of quality for 18 out of 23 proteins and was either the fastest or second fastest for 12 of the proteins). In many cases, ANC-local produces significantly higher quality results than the other methods. ANC-local removes the burden of deciding which method to use, leverages the strengths of the individual input methods, and it is extendable to include other future connection methods.

References

[1] Ibrahim Al-Bluwi, Marc Vaisset, Thierry Siméon, and Juan Cortés. Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods. *BMC Structural Biology*, 13(1), 2013.

[2] N. M. Amato and G. Song. Using motion planning to study protein folding pathways. *J. Comput. Biol.*, 9(2):149–168, 2002. Special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2001.

[3] 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. Int. Conf. Comput. Molecular Biology (RECOMB)*, pages 12–21, 2002.

[4] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.

[5] 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(1):235–242, 2000.

[6] Joseph D. Bryngelson and Peter G. Wolynes. Spin glasses and the statistical mechanics of protein folding. *Proc. Natl. Acad. Sci. USA*, 84:7524–7528, 1987.

[7] F. Chiti and C. M. Dobson. Protein misfolding, functional amyloid, and human disease. *Annu. Rev. Biochem.*, 75:333–366, 2006.

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

[9] David G. Covell. Folding protein α-carbon chains into compact forms by Monte Carlo methods. *Proteins: Struct. Funct. Bioinf.*, 14(3):409–420, 1992.

[10] W. A. Eaton, V. Muñoz, P. A. Thompson, C. Chan, and J. Hofrichter. Submillisecond kinetics of protein folding. *Curr. Op. Str. Biol.*, 7:10–14, 1997.

[11] Chinwe Ekenna, Sam Ade Jacobs, Shawna Thomas, and Nancy M. Amato. Adaptive neighbor connection for PRMs: A natural fit for heterogeneous environments and parallelism. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 1–8, Tokyo, Japan, November 2013.

[12] Chinwe Ekenna, Shawna Thomas, and Nancy M. Amato. Adaptive neighbor connection aids protein motion modeling. In *RSS Workshop on Robotics Methods for Structural and Dynamic Modeling of Molecular Systems (RMMS)*, July 2014.

[13] Chinwe Ekenna, Shawna Thomas, and Nancy M. Amato. Improved roadmap connection via local learning for sampling based planners. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, September 2015.

[14] Viara P. Grantcharova, David S. Riddle, Jed V. Santiago, and David Baker. Important role of hydrogen bonds in structurally polarized transition state folding of the src SH3 domain. *Nat. Struct. Biol.*, 5(8):714–720, 1998.

[15] D. Hsu, G. Sánchez-Ante, and Z. Sun. Hybrid PRM sampling with a cost-sensitive adaptive strategy. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3885–3891, 2005.

[16] Sophie E. Jackson, Nadial elMasry, and Alan R. Fersht. Structure of the hydrophobic core in the transition state for folding of chymotrypsin inhibitor 2: A critical test of the protein engineering method of analysis. *Biochemistry*, 32:11270–11278, 1993.

[17] D. J. Jacobs. Generic rigidity in three-dimensional bond-bending networks. *J. Phys. A: Math. Gen.*, 31:6653–6668, 1998.

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

[19] John Kuszewski, G. Marius Clore, and Angela M. Gronenborn. Fast folding of a prototypic polypeptide: The immunoglobulin binding domain of streptococcal protein G. *Protein Sci.*, 3: 1945–1952, 1994.

[20] M. Levitt. A simplified representation of protein conformations for rapid simulation of protein folding. *J. Mol. Biol.*, 104:59–107, 1976.

[21] M. Levitt. Protein folding by restrained energy minimization and molecular dynamics. *J. Mol. Biol.*, 170:723–764, 1983.

[22] R. Li and C. Woodward. The hydrogen exchange core and protein folding. *Protein Sci.*, 8(8):1571–1591, 1999.

[23] T. Liu, A. W. Moore, A. Gray, and K. Yang. An investigation of practical approximate nearest neighbor algorithms. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, pages 825–832, Cambridge, Massachusetts, 2005. MIT Press.

[24] Jose C. Martínez and Luis Serrano. The folding transition state between SH3 domains is conformationally restricted and evolutionarily conserved. *Nat. Struct. Biol.*, 6(11):1010–1016, 1999.

[25] Andreas Matouschek, Luis Serrano, Elizabeth M. Meiering, Mark Bycroft, and Alan R. Fersht. The folding of an enzyme v. H/^{2}H exchange–nuclear magnetic resonance studies on the folding pathway of barnase: Complementarity to and agreement with protein engineering studies. *J. Mol. Biol.*, 224:837–845, 1992.

[26] S. Matysiak and C. Clementi. Minimalist protein model as a diagnostic tool for misfolding and aggregation. *J. Mol. Biol.*, 363(1):297–308, 2006.

[27] T. McMahon, S. Jacobs, B. Boyd, L. Tapia, and N. M. Amato. Local randomization in neighbor selection improves PRM roadmap quality. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 4441–4448, 2012.

[28] Victor Muñoz, Eric R. Henry, James Hoferichter, and William A. Eaton. A statistical mechanical model for β-hairpin kinetics. *Proc. Natl. Acad. Sci. USA*, 95(11):5872–5879, 1998.

[29] S. Nauli, B. Kuhlman, and D. Baker. Computer-based redesign of a protein folding pathway. *Nature Struct. Biol.*, 8(7):602–605, 2001.

[30] E. Plaku and L.E. Kavraki. Quantitative analysis of nearest-neighbors search in high-dimensional sampling-based motion planning. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, July 2006.

[31] David S. Riddle, Viara P. Grantcharova, Jed V. Santiago, Eric Alm, Ingo Ruczinski, and David Baker. Experiment and theory highlight role of native state topology in SH3 folding. *Nat. Struct. Biol.*, 6(11):1016–1024, 1999.

[32] G. Song, S.L. Thomas, K.A. Dill, J.M. Scholtz, and N.M. Amato. A path planning-based study of protein folding with a case study of hairpin formation in protein G and L. In *Proc. Pacific Symposium of Biocomputing (PSB)*, pages 240–251, 2003.

[33] Kaare Teilum, Birthe B. Kragelund, Jens Knudsen, and Flemming M. Poulsen. Formation of hydrogen bonds precedes the rate-limiting formation of persistent structure in the folding of ACBP. *J. Mol. Biol.*, 301:1307–1314, 2000.

[34] Shawna Thomas, Xinyu Tang, Lydia Tapia, and Nancy M. Amato. Simulating protein motions with rigidity analysis. *J. Comput. Biol.*, 14(6):839–855, 2007. Special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2006.

[35] Jeffrey K. Uhlmann. Satisfying general proximity / similarity queries with metric trees. *Information Processing Letters*, 40 (4):175 – 179, 1991.

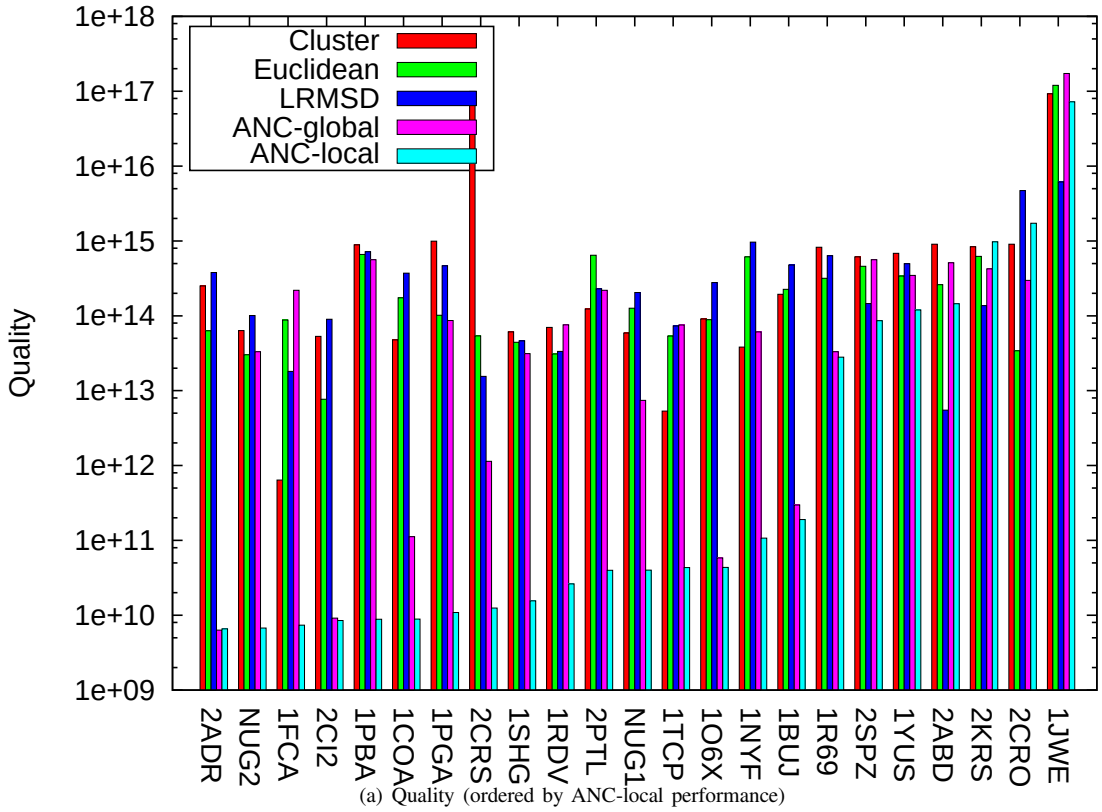
[36] Ana Rosa Viguera, Luis Serrano, and Matthias Wilmanns. Different folding transitions states may result in the same native structure. *Nat. Struct. Biol.*, 3(10):874–880, 1996.

[37] Virtudes Villegas, Jose C. Martínez, Francesc Z. Avilés, and Luis Serrano. Structure of the transition state in the folding process of human procarboxypeptidase A2 activation domain. *J. Mol. Biol.*, 283:1027–1036, 1998.

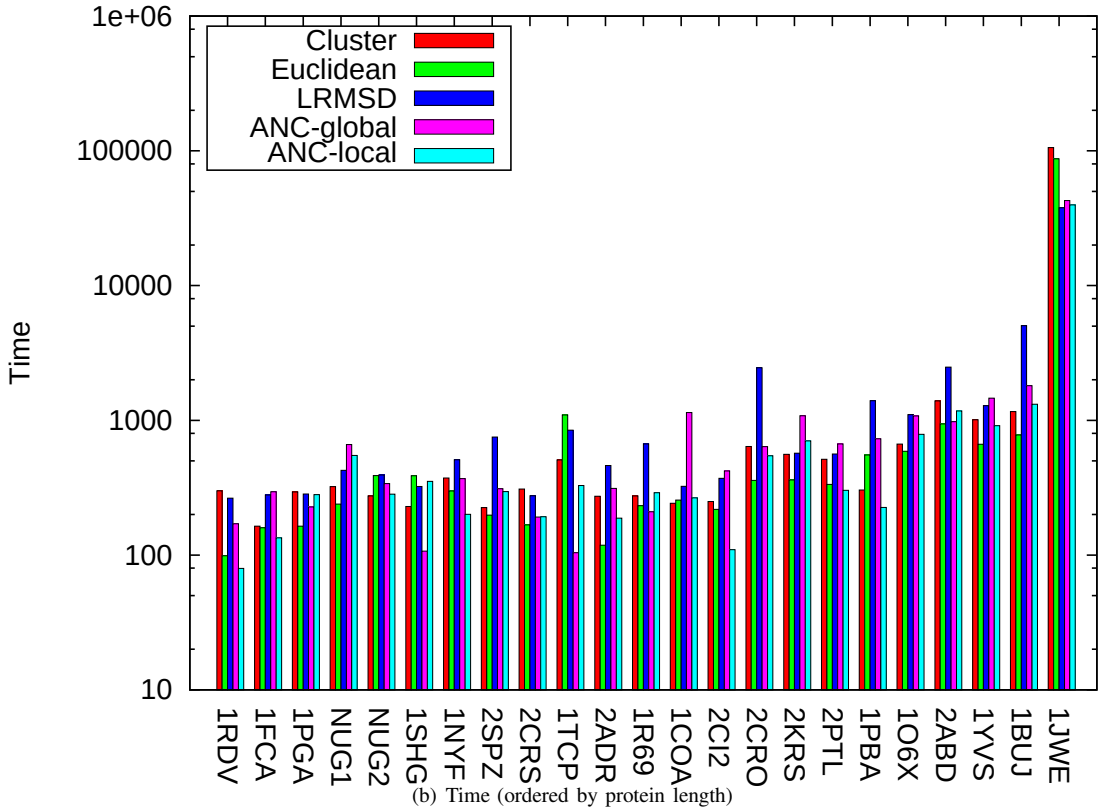
[38] Thomas E. Wales and John R. Engen. Hydrogen exchange mass spectrometry for the analysis of protein dynamics. *Mass Spec. Rev.*, 25(1):158–170, 2006.

[39] A. Yershova and S. M. LaValle. Improving motion-planning algorithms by efficient nearest-neighbor searching. *IEEE Trans. Robot. Automat.*, 23(1):151–157, 2007.

[40] Qian Yi and David Baker. Direct evidence for a two-state protein unfolding transition from hydorgen-deuterium exhchange, mass spectrometry, and NMR. *Protein Sci.*, 5:1060–1066, 1996.



(a) Quality (ordered by ANC-local performance)



(b) Time (ordered by protein length)

Fig. 2. Performance of each individual connection method, ANC-global, and ANC-local in terms of (a) roadmap quality and (b) time for all proteins studied. Note that entries are ordered by ANC-local performance in (a) and by protein length in (b).