PROBABILISTIC ALGORITHMS FOR MODELING
PROTEIN STRUCTURE AND DYNAMICS

by

Kevin P Molloy
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computer Science

Committee:

_____	Dr. Amarda Shehu, Dissertation Director
_____	Dr. Daniel Barbará, Committee Member
_____	Dr. Estela Blaisten-Barojas, Committee Member
_____	Dr. Jyh-Ming Lien, Committee Member
_____	Dr. Sanjeev Setia, Department Chair
_____	Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering
Date: _____	Spring Semester 2015 George Mason University Fairfax, VA

Probabilistic Algorithms for Modeling Protein Structure and Dynamics

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Kevin P Molloy
Master of Science
George Mason University, 2011
Bachelor of Science
George Mason University, 1998

Director: Dr. Amarda Shehu, Professor
Department of Computer Science

Spring Semester 2015
George Mason University
Fairfax, VA

Dedication

I dedicate this dissertation to my family, for their endless support and patience.

Acknowledgments

I would like to thank my advisor, Amarda Shehu, for her guidance during my studies. I would like to thank my wife, Liz, and my two daughters, Jessica and Abigail, for being very supportive and understanding. I would like to thank my parents, John and Rebekah, for their continuous support through all the trials of life.

I would like to thank my committee members for their valuable time and input. During my time at George Mason, I have been fortunate to have had some incredible professors who have challenged me. I would like to thank all of my professors for their dedication and time spent after class discussing problems and issues. I would like to thank Dr. Daniel Menascé, who convinced me I was not too old to go back to school and helped me author my first paper, Dr. Zoran Duric who always had an ear to lend when times were tough, and for Dr. Daniel Barbará for his advice and guidance.

I would like to thank Colonel Roy E. Parker for taking me to the commodore 64 user group meetings at the age of 9 and encouraging me with my early computer initiatives. I would like to thank my aunt Joanne Moll, who actually helped me debug my first assembler program while I was in 6th grade. I would like to thank my high school AP computer science professor, Lt. Col. Ken Jenkins. I would also like to thank my brother-in-law, Dr. Jack Dorminey, whose continuous support and guidance assisted me throughout my graduate studies.

I would like to thank all members of Shehu Computational Biology lab; specifically, Dr. Brian Olson for providing valuable feedback on ideas, papers, and presentations, and Dan Veltri and Rudy Clausen, for helping a computer scientist navigate the world of biology.

Table of Contents

List of Tables

List of Figures

Abstract

PROBABILISTIC ALGORITHMS FOR MODELING PROTEIN STRUCTURE AND DYNAMICS

Kevin P Molloy, PhD

George Mason University, 2015

Dissertation Director: Dr. Amarda Shehu

This thesis proposes novel probabilistic algorithms to address critical open problems in computational structural biology regarding the relationship between structure, dynamics, and function in protein molecules. The focus on protein modeling research is warranted due to the ubiquity and central role of proteins in life-critical processes in the living cell. A study of protein molecules is important for understanding our biology and health. Many disorders in the sick cell are proteinopathies, where a protein disrupts a chemical process, causing the cell to deviate from its intended biological activity. However, unlike other life-critical macromolecules, such as DNA and RNA, where significant information about activity can be extracted from knowledge of the ordering of the constitutive building blocks, proteins exhibit a more complex relationship between the order of building blocks, the structures arising from spatial arrangements of the building blocks in three-dimensional space, and the determination from such arrangements of biological activity. Since studies of proteins pose exceptional challenges in wet laboratories, the work presented in this thesis proposes powerful algorithms to complement wet-laboratory research on understanding the relationship between structure, dynamics, and function in protein molecules.

Specifically, this thesis addresses three main problems that permeate protein modeling research. The first problem, known as "from-structure-to-function," asks how to infer the function of a protein from knowledge of its active structure. The second problem, known as "from-sequence-to-structure," relates to the open question of how to predict the biologically-active structure of a protein when provided information on the identities and order of constitutive building blocks. The third problem advances the current computational treatment of proteins to alleviate assumptions of their rigidity and instead model them as dynamic macromolecules switching between structures to tune their biological activity. The objective here is to model protein dynamics efficiently by computing the molecular motions employed in structural transitions among diverse functionally-relevant states of a protein.

The algorithmic techniques employed in this thesis span machine learning, computational geometry, and stochastic optimization. In particular, we combine computational geometry and machine learning in a novel framework to infer the function of a protein from knowledge of its structure. In our treatment of the *de novo* structure prediction problem, we employ and investigate in detail an adaptive stochastic optimization framework capable of balancing between search breadth and depth in the exploration of a high-dimensional and nonlinear search space. We pursue such frameworks further and propose novel robotics-inspired probabilistic algorithms to model protein dynamics. In particular, in our treatment of structure and dynamics, we exploit analogies between protein modeling and the motion planning problem in robotics, which allow us to employ relevant concepts from motion planning algorithms and propose powerful algorithms capable of handling highly-constrained articulated systems with hundreds or thousands of continuous and discrete variables.

This thesis advances protein modeling research by extending the size and complexity of systems that can be modeled, as well as the detail and accuracy with which relevant biological questions can be answered. For instance, algorithms proposed here to model structural transitions are now able to explain the impact of sequence mutations on protein function. Just as important, the algorithmic techniques proposed in this thesis are of general utility to other domains in computer science focusing on extending optimization algorithms for vast and nonlinear search spaces of complex systems.

Chapter 1: Introduction

This thesis proposes novel algorithms to unravel the relationship between structure, dynamics, and function in protein molecules. The focus on proteins is warranted for three main reasons. First, proteins play a central role in virtually every chemical process in the living cell [6]. Second, many disorders in the sick cell are already characterized as proteinopathies, where a protein that is central to a chemical process deviates from its intended biological activity [7–10]. Third, unlike other life-critical macromolecules such as DNA and RNA, where significant information about biological activity can be extracted from knowledge of the order of the constitutive building blocks, in proteins there is a more complex relationship between the order of building blocks, their arrangement in three-dimensional space under physiological conditions, and the determination from such an arrangement of biological activity or protein function [11]. For these reasons, a study of protein molecules is both central to molecular biology and our health. More importantly, studies of proteins pose exceptional challenges both in the wet and dry laboratories. In this thesis, we focus on the computational challenges, as our goal is to propose algorithms to complement and aid wet-laboratory investigations.

Specifically, this thesis addresses three main problems that currently permeate protein modeling research in computational biology. The first problem, which we address in chapter 3, relates to the open question of how to infer what the function of a given protein is when provided information on the placement of its building blocks in three-dimensional space under physiological conditions, otherwise known as protein structure. This is often known as the "from-structure-to-function" question in computational biology, and in chapter 3 we propose a machine learning approach to address this problem. The second problem, which we address in chapter 4, relates to the open question of how to predict the structure of a protein when provided information on the identities and order of building blocks

in the protein chain. This is often known as the "*de novo* structure prediction problem" and we investigate a robotics-inspired stochastic optimization framework for its ability to balance computational efficiency and accuracy when addressing this problem. The third problem, which we address and study in detail in chapters 5, 6, and 7 advances the current computational treatment of proteins to alleviate assumptions of their rigidity. Indeed, in chapter 5, we model proteins as dynamic macromolecules, and propose a novel robotics-inspired tree-based search framework to compute motions of proteins between two distinct functionally-relevant structures. We pursue this line of investigation deeper in chapter 6, where we demonstrate the promise of combining continuous and discrete modeling in extracting information about structural transition in healthy and aberrant forms of a protein central to many human cancers. In chapter 7 we pursue further a novel algorithmic framework for the computation of structural transitions in proteins and identify both important advances and remaining challenges.

It is worth noting that the problems addressed in this thesis remain open in computational biology. More importantly, they pose interesting and challenging settings for novel algorithmic research. In this way, while the research described in this thesis is driven by specific open questions in computational and molecular biology, the algorithms described here make important contributions in computer science, as the study of biologically-realistic systems such as proteins exposes challenging systems where novel modeling and simulation algorithms need to be devised. Such a setting is unforgiving; not only do the algorithms need to be computationally efficient, but they also have to be able to perform well on realistic systems and generate data that can be trusted to make decisions. It is worth noting that computational research in macromolecular modeling research has recently gained an important place in science; all 2013 Nobel laureates in chemistry represented computational research in macromolecular modeling and simulation.

The foundation of this thesis is that protein structure determines protein function. This was demonstrated early, by Anfinsen's experiments [12]. The central role of protein structure is not surprising, as biological activity of a protein molecule is the result of binding with

small molecules or docking onto other macromolecules, including proteins. The process of binding or docking relies on strong geometrical and chemical complementarity of two molecular structures. Thus, the strong relationship between structure and function in proteins justifies a mechanistic treatment of protein molecules, under which the physiological/native three-dimensional (3d) structure of a protein determines to a great extent protein function.

In many studies focused on extracting information about the function of a protein identified in some organism, structure is seemingly circumvented. Instead, the function of an unknown target protein is often inferred from that of a known protein with a highly similar (more than 15% identical) sequence to the target. This is the basis of comparative modeling, an area of computational biology that is now well-developed and mature, greatly due to the rigor and effectiveness of dynamic programming algorithms capable of comparing two strings. In fact, nowadays, the majority of methods used for genome-wide functional annotation are based on sequence comparisons and use sequence alignment to identify homologous (ancestor-sharing) proteins. Well-known sequence alignment tools include BLAST [13], PROSITE [14, 15], and PFAM [16, 17]. These tools have become indispensable, given that genome sequencing efforts utilizing high-throughput technologies are now elucidating millions of protein-encoding sequences lacking any functional characterization [18, 19].

It is important to note that the inference of functional similarity from sequence similarity does not remove considerations of structure. Instead, two proteins of highly similar sequences have highly similar structures, and it is similarity of structures that indeed allows one to infer functional similarity. More importantly, the exquisite role of structure can be better appreciated on cases where functional similarity occurs despite low sequence similarity. Sequence-based function inference may miss detecting similar proteins where either early branching points (in such case the proteins are referred to as *remote homologs*) or convergent evolution has resulted in high sequence divergence while largely preserving structure and function. The presence of remote homologs was identified as early as 1960, when Perutz and colleagues showed through structural alignment that myoglobin and hemoglobin have similar structures but different sequences [20]. Since then, many sequence-based methods

have been offered to extend the applicability of sequence alignment tools to detect *remote homologs* [21–23]. The most successful ones, relying on statistical models learned over multiple aligned sequences, have been shown to improve upon methods based on pairwise sequence comparison but still fail to recognize remote homologs with sequence identity less than 25% [24]. It is worth noting that about 25% of all sequenced proteins are estimated to fall in this category.

In chapter 3 we advance the argument of how to infer function similarity from structure similarity for remote homologs. We proceed utilizing a structure-based method rather than a sequence-based one. Because structure is under more evolutionary pressure to be preserved than sequence, methods that compare structures allow effectively casting a wider net at detecting related proteins for functional annotation. Structure-based function inference promises to detect remote homologs and expand options for assigning function to a novel protein sequence. While many methods exist to determine whether two structures are similar, they are computationally demanding and not amenable to a high-throughput setting where a protein structure with unknown function is potentially compared against a database of protein structures with known functional annotations. One of the contributions of the work in this thesis is a novel representation of protein structure that allows expedient comparison of two protein structures. When coupled with a state-of-the-art machine learning method, this representation allows prediction of protein function from a given protein structure.

While structure-based function prediction is in principle now viable, it takes considerably more effort in the wet laboratory to elucidate structure, that is the native 3d arrangement of the amino-acid building blocks in a protein, than to determine sequence, that is the identity and order of amino acids that constitute a protein chain. There are currently no high-throughput experimental technologies for protein structure determination. While great progress is being made (for instance, as of November 2014, the Protein Data Bank [25] contains $100,000$ protein structures), the gap between known protein structures and known protein sequences has grown at an exponential rate. This is illustrated in Figure 1.1. For

this reason, computational research in protein structure determination plays an important complementary role to wet-lab technologies.

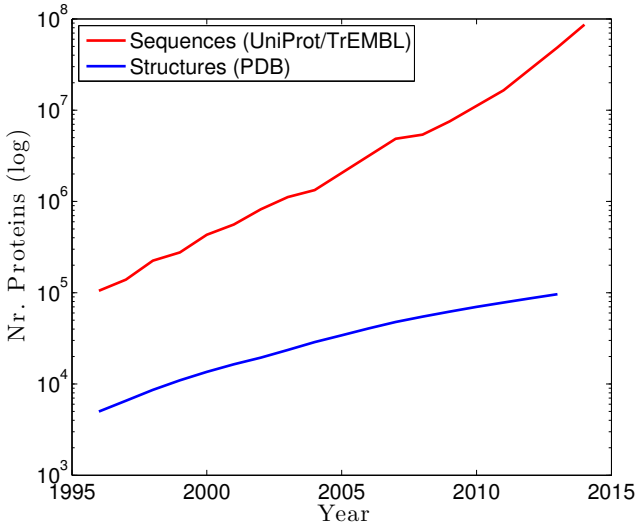


Figure 1.1: Comparing the growth in cataloged protein sequences in UniProt (red line) to determined protein structures in the PDB (blue line).

In chapter 4 we address the problem of protein structure prediction. Specifically, we address a more challenging setting and focus on proteins where sequence similarity cannot be used as a means to infer structure from a known protein to the target one (the latter is the domain of template-based modeling). Instead, we address template-free or *de novo* protein structure prediction. In particular, we approach the problem under the umbrella of stochastic optimization and focus on the analysis of novel algorithmic components to balance conflicting objectives when navigating a vast, high-dimensional space in search of lowest-energy minima possibly containing the native protein structure. A detailed treatment of this problem and our work on it is provided in chapter 4.

While the problem of *de novo* structure prediction is often characterized as the holy

grail of computational biology, it is often addressed in a somewhat simplified view of proteins. While there was early evidence from Feynmann and Schroedinger that proteins, like many physics-based systems, are not rigid but rather dynamic molecules [26, 27], we now have experimental evidence that many proteins exploit a menu of thermodynamically-stable structures through which to modulate their function and act as dynamic molecular machines [28–32]. The elucidation of series of structures that a protein uses to transition between two functionally-relevant ones, also known as a (structural) transition pathways is important not only for a detailed system's understanding but also in practical health-related settings. There are many proteins where mutations do not remove the ability of a protein to occupy functionally-relevant structures but instead modify transition pathways, making it harder or easier for a protein to transition between two or more important functional states. While the computation of such pathways has predominantly been the domain of molecular dynamic methods, such methods are typically computationally-impractical. In this thesis, we pursue an alternative approach that gains inspiration from a related problem in algorithmic robotics, known as robot motion planning.

In chapter 5 we demonstrate the ability of a novel, tree-based robotics-inspired algorithm to compute physically-realistic motions of a protein between two given functionally-relevant structures. We then pursue a more general setting and adapt roadmap-based algorithms to compute multiple paths. In particular, in chapter 6 we realize the relationship between a roadmap constructed to map the connectivity among computed low-energy structures of a protein and a markov state model. By employing Markov state theory we are able to quantify differences in transition pathways between healthy and aberrant forms of a protein central to human cancer and are thus able to obtain an explanation for how sequence mutations impact protein function. The last chapter in this thesis, chapter 7, identifies some remaining challenges and charts possible advances in this direction.

Before we begin, we relate some preliminaries in chapter 2 on protein geometry and theoretical foundations of protein biophysics that justify the computational approach pursued in this thesis. After relating our work on each of the three main problems addressed

in this thesis, we conclude in chapter 8 with some introspection and possible future work for computer science researchers interested in the challenges arising in protein modeling research.

Chapter 2: Preliminaries

This chapter outlines preliminaries on protein geometry and energetics that are essential to understanding the state of the art in protein modeling. The chapter concludes with methods for evaluating and comparing protein structures, which are essential for validation of some of the algorithms presented in this thesis.

2.1 Protein Geometry and Representation

Proteins are chains of amino acids. Each amino acid type consists of the common backbone atoms, N, C_α, C, O, and the side-chain atoms. Side-chain atoms are what differentiate the different types of amino acids. An arrangement of a protein's atoms is referred to as a *conformation*. In computational biology literature, the terms structure and conformation are routinely interchanged. However, the term conformation is more general than structure. It is the equivalent of configuration and state in system modeling research. While the structure of a protein is uniquely described by listing the cartesian coordinates of its atoms, a conformation relates more to the choice of representation of a protein chain. This may include modeling only certain atoms of each amino acids (for instance, the central C_α atom or all backbone atoms of each amino acid), whether doing so by selecting cartesian coordinates as parameters/variables of the representation or other variables (for instance, angles defined over bonds connecting atoms in a chain). The term conformation is related to that of configuration.

Small proteins can be comprised of thousands of atoms and therefore can have thousands of DOFs. To reduce the complexity, many protocols reduce the DOFs to the set of dihedral angles over the backbone atoms. This is shown in Figure 2.1. The bond lengths and bond angles are held at constant values, which is commonly referred to as the *idealized*

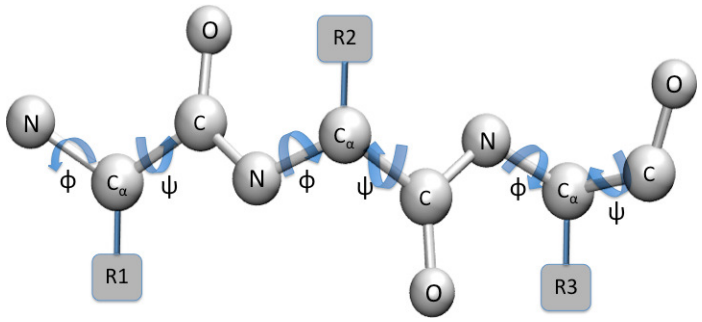


Figure 2.1: A coarse-grained protein representation where the dihedral angles, ϕ and ψ, represent the only variables or degrees of freedom (DOFs) in this model. Side-chain configurations are represent by the R1, R2 and R3 groups. This figure has been produced with the visual molecular dynamics (VMD) software [2].

representation or idealized geometry model [33]. This representation defines $2n$ DOFs for a protein consisting of n amino acids.

2.2 Protein Energy

In a thermodynamics treatment [12, 34], the sought native structure of a target protein sequence theoretically resides at the bottom of a global minimum of the protein energy surface [12]. An energy function sums the physical interactions among atoms in a protein chain and allows associating an internal energy value with a protein conformation. The protein energy surface is multi-dimensional but funnel-like, with the native state residing at the deepest minimum. Though steep, the surface is not smooth but rather rugged due to structural frustrations (that is, slight changes in structure causing large energetic jumps) [35].

It is worth noting that thermodynamics theory relates the native state of a protein to the lowest free-energy state. This state consists of a set of highly-similar structures, and free energy includes not just the average potential energy of a state but also a measure of its diversity (through the notion of entropy). However, estimating free energy is an open area of research [34–36], ripe with more inaccuracies due to additional challenges with

measuring entropy, and thus avoided by most computational treatments. Instead, as in most treatments, in this thesis the energy surface is sampled one point at a time, with a point corresponding to a conformation with an associated internal energy value. This approach necessitates that one obtain a good map of the lowest-energy regions of the energy surface before concluding where the native state resides.

Design of internal energy functions is currently an open area in computational biology and chemistry. As in system modeling, we do not have access to the energy function nature uses. We also cannot rely on quantum mechanical calculations to rigorously measure the potential energy for chains of more than 3 amino acids. Hence, all protein (and, more generally, macromolecular) modeling research relies on imperfect, semi-empirical energy functions. A detailed treatment of the computational chemistry process through which such functions are designed is beyond the scope of the work presented in this thesis. However, many studies (including our own, presented in this thesis) demonstrate that all protein energy functions have inaccuracies and often lead simplistic optimization methods to deep minima that do not correspond to the native structure of a protein [37]. This is particularly the case when employing expedient low-resolution protein representations, where low energies are associated with conformations sometimes $4-8$Å away from the known native structure of a protein sequence [38–41].

In this thesis, we plug in energy functions into algorithmic frameworks. In chapter 4 we investigate a versatile framework that is able to deal with the present inaccuracies within energy functions. In particular, the two energy functions we employ here are an in-house implementation of the Associative Memory Hamiltonian with Water (AMW), originally proposed in [42], and the open-source implementation of the Rosetta suite of functions available in the Rosetta modeling software [43]. While in some of our work we draw differences between the two, in the most recent work in this thesis we exclusively switch to the Rosetta suite of energy functions, due to speed of implementation and higher, demonstrated accuracy.

2.2.1 AMW Energy Function

The AMW function, a modification of the low-resolution potential originally proposed in [44], has been used previously by Shehu and collaborators for *de novo* structure prediction [5, 45–49]. AMW sums non-local terms (local interactions, such as bond length fluctuations, are kept at ideal values in the idealized geometry model): $E_{\mathrm{AMW}} = E_{\mathrm{Lennard-Jones}} + E_{\mathrm{H-Bond}} + E_{\mathrm{contact}} + E_{\mathrm{burial}} + E_{\mathrm{water}} + E_{\mathrm{Rg}}$. The $E_{\mathrm{Lennard-Jones}}$ term is implemented after the 12-6 Lennard-Jones potential in AMBER9 [50] allowing a soft penetration of van der Waals spheres. The $E_{\mathrm{H-Bond}}$ term allows modeling hydrogen bonds and is implemented as in [51]. The other terms, E_{contact}, E_{burial}, and E_{water}, allow formation of non-local contacts, a hydrophobic core, and water-mediated interactions, and are implemented as in [52]. The E_{Rg} favors collapse by penalizing conformations with radius of gyration significantly different from theoretically-calculated values [48].

2.2.2 Rosetta Energy Function

The Rosetta software package implements a suite of different scoring functions. In particular, a total of 6 different scoring functions are used in the low-resolution stage of the *de novo* structure prediction protocol used in Rosetta. These correspond to different assignments to the weights that measure the contribution of different local and non-local energy terms. What we refer to as the Rosetta energy function is a linear combination of all possible 10 energy terms, which measure repulsion, amino-acid propensities, residue environment, residue pair interactions, three terms measuring interactions between secondary structure elements, and three other terms measuring density and compactness of structure (cf. to Ref. [53] for more details).

The low-resolution stage in the Rosetta protocol consists of 4 different substages, each with different scoring functions. The first substage conducts 1-2 cycles of $2,000$ MMC moves each starting with an extended chain and using the score0 assignment. The only energy term modeled is a soft steric repulsion, and its purpose is to yield a random starting conformation.

The second substage of $2,000$ MMC moves uses score1 to accumulate secondary structure. The third substage uses 5 cycles of $2,000$ MMC moves each with score2 followed by a cycle of $2,000$ MMC moves with score5; score2 includes terms to favor hydrophobic collapse and beta strand pairings, whereas score5 lacks these two terms to allow relaxation. The fourth and final substage consists of 3 cycles of $4,000$ MMC moves each and uses score3, which has all the possible energy terms except for hydrogen bonding. The ensuing selection analysis in preparation for side-chain packing and energetic refinement uses score4 to rank low-resolution conformations; score4 does not have any compaction or beta-strand pairing terms.

In light of this intricate protocol of different scoring functions, what we refer to as the Rosetta energy function in this thesis is score3, as this is the one that has the highest number of Rosetta energy terms in the low-resolution stage, and all other scoring function in the low-resolution stage can be viewed as a scaled variant of score3. In some of the experiments in this thesis we focus on the evaluation of high-resolution, all-atom models. In such cases, we employ the score12 Rosetta function.

It is worth noting that like most energy functions, the AMW and Rosetta suite of functions are evaluated over cartesian coordinates of modeled atoms. When the representation chosen for a protein is angular-based, as is done in this thesis, an additional step is needed to compute cartesian coordinates from values of angles. This is a well-understood step known as forward kinematics in computational geometry and is linear in the number of angles [54]. The computational cost of evaluating an energy function is high, as the most expensive term in such functions is often the one summing up interactions among non-bonded atoms. This term is responsible for the quadratic time complexity of energy function evaluations.

2.3 Measurements: Comparing Protein Structures and Conformations

Comparing two protein structures is a well-studied but open problem. The difficulty resides in designing a distance function that accurately captures intrinsic differences between two protein structures. This is not an easy task, as it generally relates to the problem of designing distance functions for a high-dimensional space. However, in this thesis we use a baseline, well-understood dissimilarity function to compare two protein structures. When dealing with protein conformations that specify angular rather than cartesian DOFs, we employ metrics such as the L1 norm.

2.3.1 Least Root Mean Square Deviation – lRMSD

One of the main measurements used in this thesis is lRMSD, which is the weighted Euclidean distance between corresponding atoms after optimal superposition of two conformations under comparison, as shown in Equation 2.1. The optimal superposition refers to the rigid-body motion or transformation in SE(3) minimizing the weighted Euclidean distance [55]. lRMSD captures structural dissimilarity but is not a Euclidean metric, as it does not obey the triangle inequality. Low values indicate high similarity, and high values indicate high dissimilarity, but interpretation of intermediate values is difficult and the subject of many studies [56]. For instance, lRMSD has been found to depend on system size. A 5Å lRMSD between a computed conformation and the native structure of a short protein chain of no more than 30 amino acids is considered a large deviation, but the same dissimilarity is less significant for a protein of 70 amino acids or more. In general, if the lowest lRMSD obtained over computed conformations to the known native structure is more than 6Å, the native structure is not considered to have been captured.

$$lRMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\vec{x} - \mathbf{U}\vec{y})^2} \tag{2.1}$$

In this thesis we also use lRMSD as a progress coordinate, ΔR, to track the distance between two conformations, as done in other works [32, 57]. ΔR is shown in Equation 2.2 and is specifically used in the context of motion computation.

$$\Delta R = lRMSD(C, C_{start}) - lRMSD(C, C_{goal}) \tag{2.2}$$

2.3.2 Global Distance Test Total Score– GDT_TS

High values of lRMSD do not necessarily indicate significant structural dissimilarity. Since lRMSD weighs each atom equally, it overly penalizes cases where differences are localized to a specific region, say a loop in different orientations in the two conformations under comparison. In such cases, other measurements, such as GDT_TS (Global Distance Test Total Score), can be more appropriate. GDT_TS essentially locates a maximum subset of atoms between two conformations under comparison that are close in space after optimal superposition and minimizes an overall lRMSD-based error. GDT_TS is reported in % and captures similarity, so higher values are better. As employed in CASP, GDT_TS = (GDT_P_1 + GDT_P_2 + GDT_P_4 + GDT_P_8)/4, where GDT_P_d is the fraction of maximum amino-acid subsets in a conformation superimposing on the reference (native, in our comparisons) structure with an lRMSD $\leq d$Å. Some of our detailed analysis below employs GDT_TS scores in addition to lRMSD.

Angular Distance Functions

The coarse-grained, angular-based representation of proteins allows for angular differences to be calculated. Proteins are represented as vectors of dihedral angles, from which the L1 norm can easily be computed. These measurements are performed in the configuration/conformational space of the protein, which has a non-linear relationship to the cartesian/work space. This presents challenges in interpretation of such distances, but one of the benefits is the linear computational cost in the number of angles compared.

Chapter 3: From Protein Structure to Protein Function

The work described in this chapter is based on preliminary work published in a conference proceeding [58] and an extended version published in a journal article [59]. In summary, we build here over fragment-based structural representations that have been proposed that allow fast detection of remote homologs with reasonable accuracy. We propose higher-order topic-based representations of protein structures, obtained through the Latent Dirichlet Allocation (LDA) model, to provide an alternative route for remote homology detection and organization of the protein structure space in few dimensions. Various techniques based on natural language processing are additionally proposed and employed to aid analysis of topics in the protein structure domain. We show that the topic-based representation is effective; we conduct a detailed analysis of the information content in the topic-based representation, showing that topics have semantic meaning. Finally, the fragment-based and topic-based representations are shown to allow prediction of superfamily membership, thus allowing prediction of function from structure. We focus in this chapter on the methodological novelty and relate representative results. The images used in this chapter are copyright of the BMC Bioinformatics Journal.

3.1 Background and Related Work on Fast Protein Structure Comparison for Functional Annotation

Work on structure comparison methods has been spurred due to the Structural Genomics Initiative [60] aiming to determine representative structures of all protein families. Such research remains challenging, mainly because the problem of finding the optimal structure similarity score is ill-posed and has no unique answer [61]. While ultimately the purpose is to transfer functional similarity to structurally-similar proteins, it remains unclear how

biologically-significant a particular structural alignment is [62, 63]. Well-known methods measuring similarity of two protein structures include those based on Dynamic Programming (DP) [64–66], including SSAP [67] and STRUCTAL/LSQMAN [68–70], methods based on distance matrices, such as DALI [71], those based on extensions of an alignment pinned at aligned fragment pairs or groups of residues, such as CE [72], LGA [73], TMAlign [74], methods based on comparisons of secondary structure units, such as VAST [75, 76] and SSM [77], and those based on comparisons of backbone fragments [78]. The majority of these methods are computationally demanding, as they rely on aligning the two protein structures provided for comparison. This is not effective in settings where such methods are intended to be employed as filters; that is, compare a protein structure against structures with known functions to identify those of similar structure.

Most filter approaches for structure comparison rely on finding suitable representations of protein structure so that fast distance measurements can be employed over the representations to rapidly score the similarity of two protein structures without the computationally-intensive step of aligning two structures under comparison [78–86]. The representations are typically string or vector-based, and characters or elements are drawn over a pre-compiled alphabet or library of structural features. Representative filter methods include SGM [87], PRIDE [88], and that in [78]. In particular, fragment-based representations of protein structures have been recently proposed to allow fast detection of remote homologs with reasonable accuracy [78]. The representations are based on the bag-of-word (BOW) model of text documents, representing a protein structure as a bag of backbone fragments. Essentially, a representative set of backbone fragments of a given length are compiled over known protein structures [89]. A protein structure of interest is then represented as a vector whose entries record the number of times each of the fragments in the compiled library of fragments approximates a segment in the given protein backbone. The resulting *fragbag* representation has been shown efficient and effective at identifying structural neighbors of a given protein, including close and remote homologs [78]. It is worth noting that fragment-based representations have also been used for structural alignments [90, 91].

We build here over the fragbag representation. The fragbag representation is based on the Kolodny fragment libraries [89] and is based on the concept of a C_α-based molecular fragment. A library of fragments of l_f amino acids in [89] is constructed as follows. Fragments of C_α traces of 200 accurately-determined protein structures are clustered, depositing the representative of each cluster in the fragment library. While analysis on the fragbag representation considers libraries of fragments of length $l_f \in \{6, \ldots, 12\}$, we focus on fragments of length 11 in this paper, shown to have the highest accuracy in identifying structural neighbors in [4, 78] and in our own analysis (data not shown).

The concept of molecular fragments allows obtaining a vector-based representation of a protein structure as follows. Given a fragment library of N fragments of a fixed length l_f, a protein structure P can be represented as a vector V of N entries. Different information retrieval (IR) techniques can be used to fill an entry V_i associated with fragment f_i in the library ($1 \leq i \leq N$). For instance, entry V_i can record the presence or absence of fragment f_i (stored at position $1 \leq i \leq N$ in the library) in P, effectively resulting in a boolean vector. Alternatively, the number of times fragment f_i is found in P can be used. This is also known as term frequency (TR), and results in what is introduced as the *fragbag* representation in [78]. Generally, other naive vector space models can be used, including term frequency-inverse document frequency (TF-IDF) [92].

The presence of a fragment f_i in P is detected as follows. The C_α trace of P (that is, only C_α coordinates are extracted from the protein structure) is inspected at every location j in blocks of f consecutive amino acids, or segments $[j, j + f - 1]$. The C_α coordinates of the particular segment under consideration are compared to each fragment f_i in the library ($1 \leq i \leq N$), and the fragment with the lowest least-root-mean-squared-deviation (lRMSD) is reported as the fragment matching the particular segment (least in lRMSD stands for optimal RMSD after removing deviations due to rigid-body motions, where lRMSD is Euclidean distance weighted over number of points) [55]. The process is illustrated in Figure 3.1.

Given the fragbag representation, any distance or similarity measurements can be used

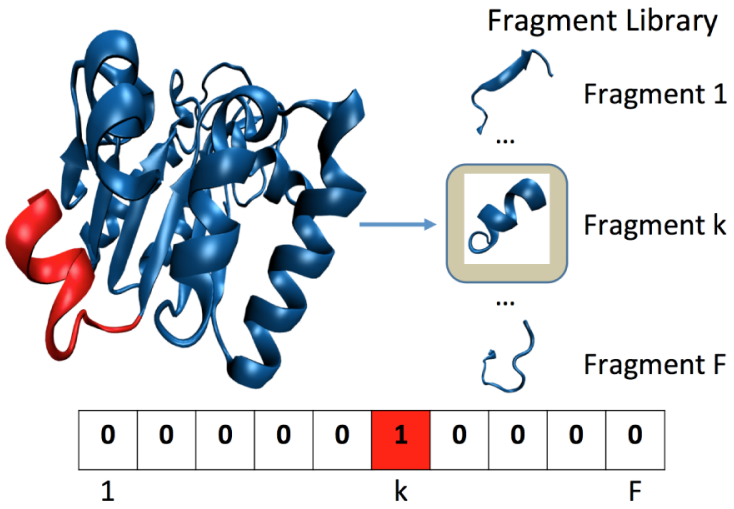


Figure 3.1: A protein structure is shown on the left, rendered with VMD [2] using the NewCartoon graphical representation. The protein structure is scanned one fragment at a time from the N- to the C-terminus. The first fragment is highlighted in red. The position of the fragment in the fragment library is identified, and the entry in the BOW vector at that particular position is incremented. After the entire structure is scanned, the resulting BOW vector is the one supplied to LDA.

over the fragbag vectors of two protein structures to measure their structural distance or similarity. In [78], various distance measurements are tested, including the basic Euclidean distance and other ones, such as cosine distance that measures the angle between two vectors. The cosine distance is reported to be most accurate and competitive with top structure-alignment methods in detecting structural neighbors. More interestingly, the entire protein structure space, as collected in the SCOP database, can be visualized, by subjecting such fragbag-represented protein structures to dimensionality reduction techniques, such as Principal Component Analysis (PCA) [93].

3.2 Method

We build here over the fragbag representation to design topic-based representations of proteins, employing LDA. The LDA model is summarized next, with further description of the topic-based representations it offers on proteins and the measurements used to conduct the

analysis over topics.

3.2.1 LDA model

At its core, LDA is a three-level hierarchical Bayesian model. As illustrated in Figure 3.2, LDA operates as follows. First, a multinomial distribution ϕ_Z is selected for each topic Z from a Dirichlet distribution with input parameter β. Second, for each protein P, a multinomial distribution θ_P is selected from a Dirichlet distribution with input parameter α. For each fragment f_i in a protein structure P, a topic $Z \in T$ is selected from the multinomial distribution θ_P. The number of topics T is specified a priori. Finally, a fragment f_i is selected from the multinomial distribution ϕ_Z.

Given P proteins, T topics, and N fragments, one can represent $p(f_i|z)$ for the fragment f_i, with a set of T multinomial distributions ϕ over N fragments, $P(f_i|Z = j) = \phi_{f_i}^{(j)}$. $P(z)$ is modeled with a set of P multinomial distributions θ over T topics. LDA assumes a prior distribution of θ and ϕ to provide a complete generative model. A Dirichlet distribution is used to choose priors α for θ and β for ϕ. We use Gibbs sampling [94] to estimate ϕ and θ and model each protein as a probability distribution over latent topics discovered by LDA. Pseudocode is provided in Algorithm 3.1 along with a visual illustration of the LDA plate in Figure 3.2.

Algorithm 3.1 The generative model used to build a new protein.

Input:
$\phi_1, ..., \phi_T$, Each topics distribution of fragments
P_{size}, Number of AA in protein
Output: Protein P

1: θ = DrawMultinomial(α) ▷ Distributions of Topics for this Protein
2: **for** pos=1,...,P_{size} - fragmentSize + 1 **do**
3: topic = SampleMultinominal(θ) ▷ Select topic for this fragment
4: fragment = SampleMultinomal(ϕ_{topic}) ▷ Select a fragment within this topic
5: P_{pos} = fragment
6: **end for**

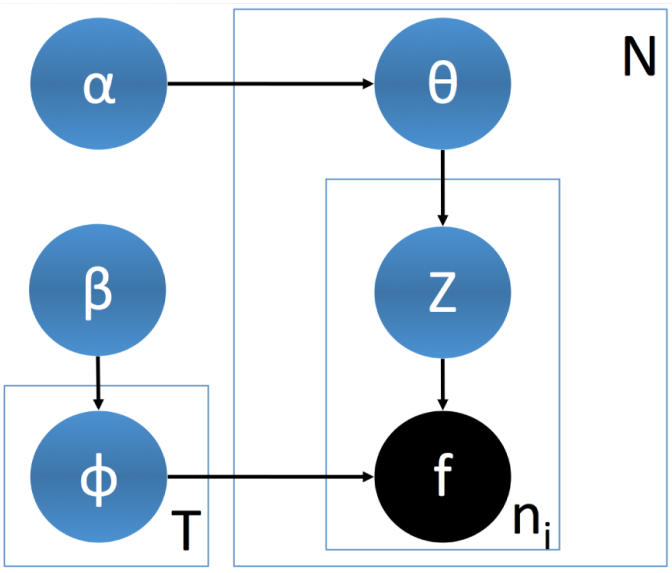


Figure 3.2: Plate diagram for LDA. T is the number of topics, N is the number of protein structures. Each fragment within a protein is represented by f and n_i is the number of fragments in P_i. Blue and black backgrounds indicate latent and observed variables respectively.

LDA-obtained topics make for general representations of proteins, under which a protein is treated as a mixture of many topics, albeit with different probabilities. One can employ these topic-based representations to identify structural neighbors of a protein. Topics can also be used to categorize the protein structure space, revealing interesting insight into what it is that each topic captures about protein structure and function.

Evaluating information content in topics

The distribution of fragments over the entire protein structure space, as available in the SCOP database, for instance, can be used to represent a baseline distribution over fragments. Each topic obtained by LDA is a probability distribution over fragments. The information gain of each topic can be measured over the baseline distribution. We use the symmetric Kullback-Leibler (KL) divergence [95] to measure the information gain of each topic over the baseline distribution. Briefly, given two probability distributions p_0

and p_1, $\mathrm{KL}(p_0,\ p_1) = \sum p_0(x) \cdot ln\frac{p_0(x)}{p_1(x)}$. We use a symmetric version of KL defined as $0.5 \cdot (\mathrm{KL}(\mathrm{p_0}, \mathrm{p_1}) + \mathrm{KL}(\mathrm{p_1}, \mathrm{p_0}))$. Larger distances imply higher information gain in each topic as opposed to the baseline distribution of fragments over the entire corpora. This evaluation is carried out for each topic in the Results section to additionally measure the information gain as one increases the number of topics requested from LDA.

In addition, log likelihood can be used to evaluate how well the data (the fragments defining protein domains) fits the model, which in this case is the topic space model produced by LDA. When performing parameter estimation, a common strategy is to maximize the log likelihood. We employ this technique to measure the effectiveness of each LDA model, varying the number of topics as before. The equation for calculating the log likelihood for each protein is: $log\ p(P_i|\mathcal{M}) = \sum_{j=1}^{F} n_i^{(j)} log(\sum_{k=1}^{T} (\ p(f_j|t_k)p(t_k|P_i)\ ))$. F is the total number of fragments used to describe the ensemble. $\mathcal{M}$ represents all the terms of the LDA model (including the number of topics). The term $n_i^{(j)}$ represents the number of times fragment j appears in protein P_i. The term $p(f_j|t_k)$ is the probability of the fragment f_j being in topic t_k and $p(t_k|P_i)$ is the probability of topic t_k being in protein P_i. These measurements are shown in the Results section to show that log likelihood decreases as the number of topics increase.

3.2.2 Topic signatures of structural classes and co-localization in protein structure space

Each topic captures "signatures" associated with different classifications (SCOP, CATH). To test for these signatures, we propose using heatmaps constructed over the LDA-computed topic space, as interpretation of topics is more challenging in non-text domain applications of LDA. LDA presents the topic space as a P x T matrix, where P is the number of proteins and T is the number of topics. The row vector for protein P_i records the number of times a fragment is classified to be within a given topic T_j. Additionally, each protein is assigned a label according to some classification standard; a label corresponds to a class. For instance,

a label may be the fold of the protein, as obtained from the top level of the SCOP hierarchy. Alternatively, the label can track the superfamily membership of a protein in SCOP.

Many protein domains are assigned the same label L_i. We sum fragment counts for topic T_j on each protein assigned the same label L_i. This provides us with a fragment count for topic T_j in label L_i. Normalizing over all labels provides us with probability $P(L_i|T_j)$. This produces an LxT matrix, where each column in the matrix sums to one. Results in this paper visualize this matrix as a heatmap, with colors following the low-to-high probabilities in a blue-to-red colors scheme.

When protein classes have strikingly different sizes, the above analysis will be skewed. A high probability $P(L_i|T_j)$ may be assigned to a class with label L_i simply because of the high number of domains in the class with label L_i. This situation arises when analyzing topic signatures over the superfamily classification in SCOP. In this case, we take a different approach to obtaining a heatmap that elucidates topic signatures for protein classes. We employ the ChiSquare significance test [96] at a confidence level of 99%. This analysis is performed for each topic T_j. For each protein with label L_i, we compute the number of fragments found within topic T_j (let's refer to this as $C_{T_j}^{L_i}$), and the number of fragments that are not assigned to proteins with this label ($C_{T_j}^{\neg L_i}$). We compute these counts for the entire population minus the topic we are currently analyzing ($C_{\neg T_j}^{L_i}$ and $C_{\neg T_j}^{\neg L_i}$). These value are used to construct a contingency table and perform the ChiSquare significance test. When the test shows a significant difference, and the population in the topic is greater than the remainder of the population, we characterize this topic as having a signature for the label under consideration.

3.2.3 Predicting superfamily membership of protein structure

We demonstrate that the fragbag and topic-based representations can be employed by machine learning classification algorithms to predict superfamily membership for a given protein structure. Since this is a multiclass classification problem, we employ the one-vs-all

strategy, using 7 binary classifiers, one for each of the 7 most-populated superfamilies in SCOP. We employe the popular Support Vector Machines (SVM) for the binary classifier [97].

The set of $9,852$ protein domains in these superfamilies is extracted, and LDA is applied to this set. When using the topic-based representation, each protein's multinomial distribution across the topic space returned by LDA serves as its coordinates in the 10-dimensional space (our analysis in the Results section makes the case that no more than 10 topics are needed). The resulting 10-dimensional vectors are treated as a training dataset, and 7 classifiers are built (SVM is a binary classifier) in order to predict superfamily membership with binary classifiers. When using the fragbag representation, the training vectors are 400-dimensional as opposed to 10-dimensional when using topics.

When building an SVM classifier for superfamily i ($1 \leq i \leq 7$), the set of training vectors corresponding to domains in that superfamily are treated as the positive training dataset. The rest of the vectors, corresponding to domains in other superfamilies are treated as the negative training dataset. We note that for some of the superfamilies, there are many more negative instances than positive ones, as expected. In such cases, re-balancing of data is performed by undersampling the negative class in order to achieve an equal count of positive and negative instances.

Each SVM classifier is trained independently (on each superfamily), using a polynomial kernel and a soft margin parameter $C = 0.1$. Ten-fold cross-validation is used to measure the classification performance. For each protein domain, the prediction among the 7 classifiers that has the highest confidence is chosen as the final prediction for that domain. In this way, superfamily membership is predicted for each family, and standard TPR, FPR, and accuracy measurements can be used to evaluate performance.

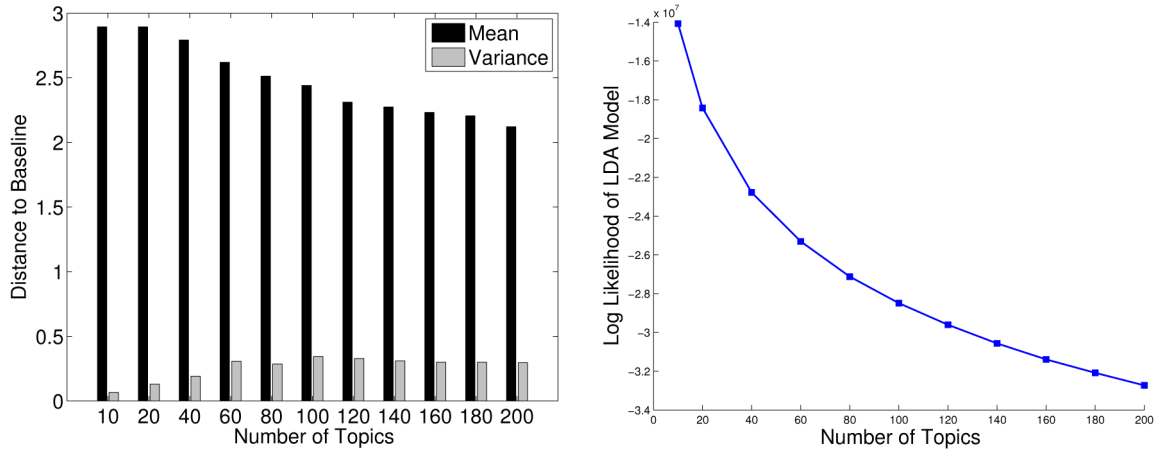
3.3 Results

We relate here representative results that make the case the topic-based representation is both meaningful and effective at predicting the function of a protein from knowledge of its structure. We employ a MATLAB implementation of LDA [98] utilizing the recommended defaults where $\alpha = 50/(\text{number of topics})$ and $\beta = 200/(\text{fragment library size})$. We utilize a test dataset containing 31,155 protein domains. Building the fragbag representation for this dataset takes 10 hours. LDA execution times are highly dependent on the number of topics, and vary from 2 hours for 10 topics up to 24 hours for 200 topics. We utilize the WEKA package for solving the SVM models used in superfamily classification [99]

3.3.1 Determining Number of Topics

Figure 3.3(a) relates the results of the procedure detailed in Methods to determine the optimal number of topics. As the number of topics increases, the symmetric KL distances decrease, suggesting that increasing the number of topics does not result in more information. The log likelihood is shown in Figure 3.3(b). As the number of topics increases, the log likelihood decreases. Many topics are essentially "junk" topics. These two measures at 11 distinct LDA models where the number of topics varies from 10 to 200 allows concluding that 10 topics is sufficient.

Thus, for the rest of the analysis presented in this chapter, a protein structure is represented as a 10-dimensional vector (where each entry in the vector records the probability with which each topic is "found" in the structure). This is in contrast to the higher-dimensional vector space resulting from the fragbag representation where 400 fragments are employed as opposed to 10 topics. One of the advantages of this lower dimensionality is that dimensionality reduction techniques do not have to be used in order to provide low-dimensional user-friendly embeddings or maps of protein structure space. A component of our analysis below illustrates how topics are signatures of SCOP classes and can even be employed to accurately predict superfamily membership.



(a) (b)

Figure 3.3: (a) Symmetric KL distances are measured between each topic and the baseline fragment distribution at 11 settings, varying the number of topics from 10 to 200. Mean and variance is shown for each setting. (b) The log likelihood of fitting the data is shown for each of the 11 LDA models.

3.4 Comparing Fragbag to Topic-based Representation

Employing the fragbag or topic-based representation and the cosine distance over the particular representation under investigation and continuously varying the decision threshold (that is, the cosine distance between two protein structures under the particular representation), a receiver operating curve (ROC) can be constructed, and the average area under the curve (AUC) score can be reported. The ROC curve plots the true positive rate (TPR = TP/(TP+FN)) vs. the false positive rate (FPR = FP/(FP+TN)) over the decision threshold. Summarizing the ROC with AUC allows associating a score with each query protein. Averaging over all proteins in the dataset, essentially treating each of them in turn as a query protein, allows obtaining an average AUC and thus measuring the effectiveness of a particular representation at capturing structural neighbors. Performing this analysis at the three different SAS thresholds further allows judging the effectiveness at capturing close to remote homologs.

Figure 3.4 compares the average AUCs obtained under each representation and additionally places them in a larger context by comparing them to two methods, SSM [77], representative of alignment-based methods, and SGM, a representative of filter methods [87]. The average AUCs reported for these methods are obtained as published in [3]. Additionally, we include the average AUCs obtained over 10 topics as reported in [4]. Figure 3.4 shows that SSM is the best performer, followed closely by fragbag and the rest. LDA and SGM are comparable.

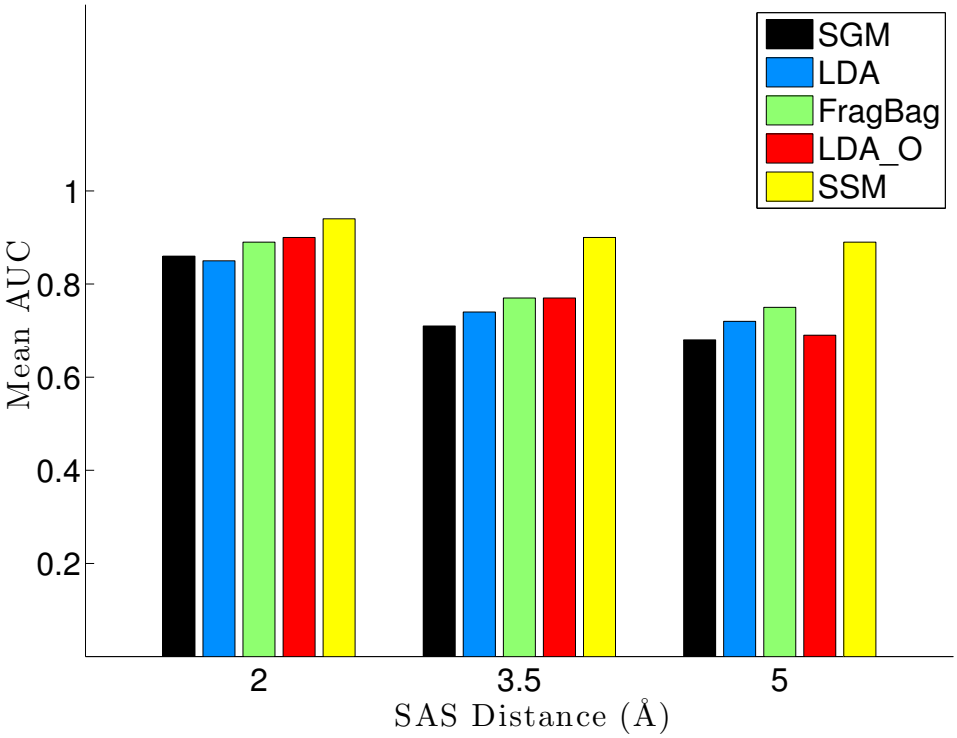


Figure 3.4: The average AUCs over the SCOP dataset, calculated as described in the Results section, are compared among different methods. Data from the SGM and SSM methods are obtained as published in [3]. These two methods are compared against the fragbag and two topic-based representations (as shown here (LDA) and in [4] (LDA_O)).

In particular, the average AUCs on each SAS threshold obtained with the fragbag and

topic-based representations are listed in Table 3.1 for a direct comparison. Two observations can be drawn. First, both representations, fragbag and topic-based, are equally effective at capturing structural neighbors at each of the three SAS thresholds. Second, under each representation, the effectiveness is higher at lower SAS thresholds (above 0.8 at a SAS threshold of 2.0Å), allowing us to conclude that the representations have an easier time capturing close homologs than remote homologs. However, performance on remote homologs remains good (higher than 0.7 at a SAS threshold of 5Å). Taken together, this experiment allows concluding that the topic-based representation allows capturing structural similarity and can be employed to rapidly extract structural neighbors (close and remote homologs) of a given protein with known structure.

Table 3.1: Avg. AUCs over frabag vs. topic-based representations.

	5Å	3Å	2.5Å
Fragbag [78]	0.75	0.77	0.89
Topic-based (this work)	0.72	0.74	0.85

3.4.1 Topic Interpretation

Inspection of the top-populated fragment and of heatmaps computed as described above allow associating a meaning with each topic. The top-populated fragments in each topic are shown in Figure 3.5.

The heatmap shown in Figure 3.6(a) color-codes topics per class at the fold level of the SCOP hierarchy in a blue-to-red color scheme tracking low-to-high probabilities measured as detailed in Methods. The results suggest that topics $1-4$ are over-represented in the α class but under-represented in the β class. This is reversed for topics $5-10$. In contrast, the other classes either have a high mixture or a low mixture of each topic. Correlating these results with the top-populated fragments provides an explanation for why this is the case. Topics

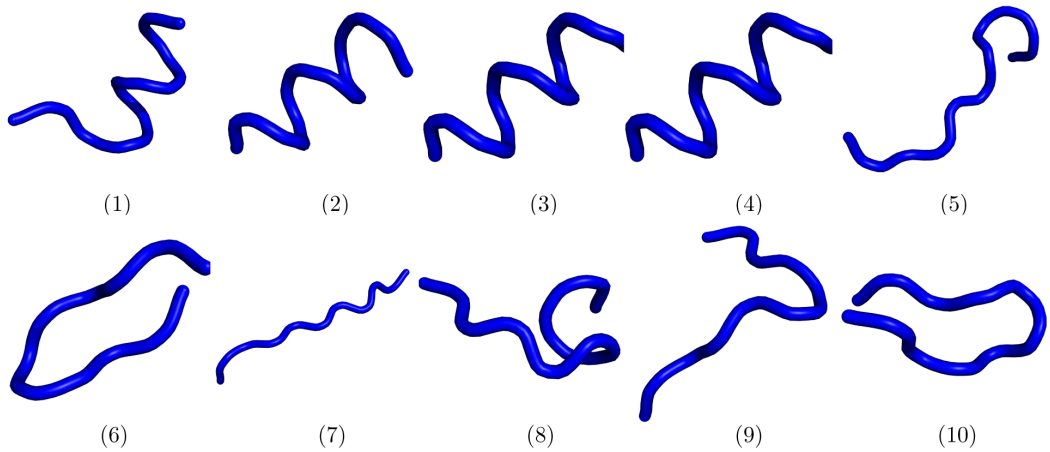


Figure 3.5: The top-populated fragment of each topic is shown here in NewCartoon representation generated using the PyMol rendering software [100].

$1-4$ are related to α-helical topologies, as evidenced by the top fragment shown. Topics $5-10$ are related instead to β-sheet topologies. Put together, these results demonstrate that classes at the fold level of the SCOP hierarchy have unique topic signatures. It is worth emphasizing that this result is made even stronger when considering that, often, domains assigned to the β class may contain a few α-helices (data not shown). The analysis suggests that topics capture structural categorization.

The heatmap shown in Figure 3.6(b) color-codes topics per class at the superfamily level, correcting for the high variance in population sizes of top superfamilies in SCOP. Blue indicates low presence of a topic, and red indicates high presence. The results suggest that superfamilies have unique topic signatures. For instance, the immunoglobulin domain has many of topics $5-10$ overrepresented. This is encouraging, as inspection of these topics reveals that they are high in β-sheet, and immunoglobulin domains are all-*beta* proteins. On the other hand, the P-loop Binding domain is rich in α-helices. Encouragingly, the topics that are overrepresented in this superfamily are topics $1-4$, which capture α-helical fragments, as shown in Figure 9. The winged helix DNA-binding domain is significantly represented in topics 1 and 3, both having high concentration of α-helical fragments. This

agrees with the SCOP classification of this domain as all α. Similarly, EF-hand is only significantly represented in topic 1, which is dominated by α-helical fragments. This agreement with the all α SCOP classification. The topic signatures capture the other superfamilies, as well, suggesting that topics additionally capture functional categorization.

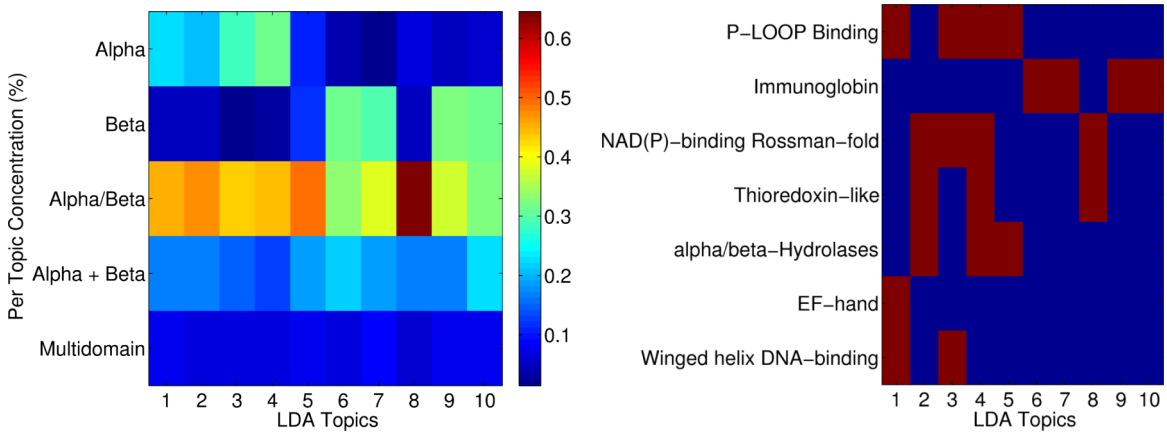


Figure 3.6: (a) Heatmaps highlight "signature" topics per class in the (a) fold level vs. (b) superfamily level of the SCOP hierarchy. Blue-to-red color scheme tracks low-to-high probabilities.

3.4.2 Predicting Superfamily Membership

A set of 7 classifiers is built as described in section 3.2.3. This experiment is repeated twice, once using the fragbag and the other using the topic-based representation. The distribution of the protein domains employed as training data in each case across the 7 superfamilies is shown in Figure 3.7. The performance of each of the 7 SVM classifiers in 10-fold validation is shown in Table 3.2. Very high accuracy ($> 80\%$), TPR (> 0.8), AUC (> 0.83), and low FPR (< 0.3) are obtained on each superfamily whether using fragbag or the topic-based representation. The fragbag representation allows for slightly better classification performance. These results confirm that the topic-based representation, while

only 10-dimensional as compared to the 400-dimensional fragbag representation, can be used to build effective classifiers of proteins, even at the superfamily level of detail.

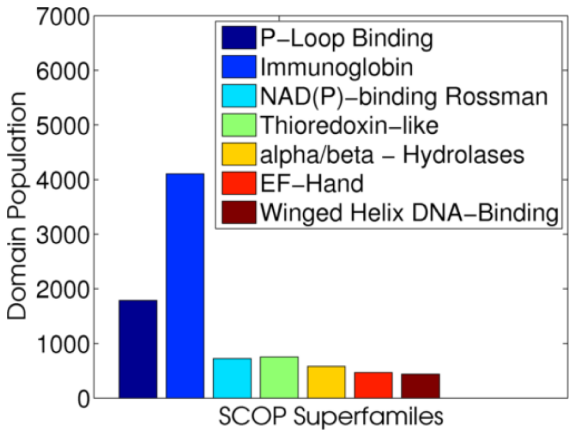


Figure 3.7: The distribution per superfamily is shown for the protein domains in the 7 most-populated superfamilies in SCOP. These domains are treated as training data for SVMs to classify proteins by superfamily.

3.5 Conclusions

The presented analysis demonstrates that fragbag and LDA-obtained topic-based representations allow capturing structural similarity. In addition, the topics are meaningful and effective at providing functional annotations in terms of superfamily membership.

Table 3.2: Performance is reported for the 7 SVM classifiers on identifying a protein domain as being a member of one of the seven SCOP superfamilies. Accuracy is the sum of true positives and true negatives divided by the number of samples. Reported values are rounded to the nearest tenth of a percent.

	Fragbag Representation				Topic-Based Representation			
SCOP Superfamily	Accuracy (%)	TPR	FPR	AUC	Accuracy (%)	TPR	FPR	AUC
P-Loop Binding	96.4	0.98	0.05	0.95	84.3	0.97	0.29	0.84
Immunoglobin	100.0	1.00	0.00	1.000	99.9	0.99	0.0	1.0
NAD(P)-binding Rossman Fold	98.7	0.99	0.02	0.99	90.9	0.94	0.13	0.91
Thioredoxin-like	98.8	0.98	0.01	0.99	80.2	0.92	0.32	0.80
alpha/beta Hydrolases	99.1	1.00	0.02	0.99	92.7	0.95	0.10	0.93
EF-hand	100.0	1.00	0.00	1.000	98.8	0.99	0.01	0.99
Winged helix DNA-binding	98.7	0.98	0.01	0.99	84.4	0.79	0.11	0.84

Chapter 4: Protein Structure Prediction Employing Robotic Methods

The work described in this chapter is based on preliminary work published in a conference proceeding [101] and an extended version published in a journal article [102]. In summary, we investigate here various algorithmic components of a robotics-inspired tree-based framework originally proposed in [5, 45] for the *de novo* structure prediction problem. In particular, we focus on the impact that biasing the search towards low-energy conformations has on adequate coverage of the conformational space. We propose different biasing strategies to steer the search towards diverse low-energy conformations while not exploiting artifacts of a given energy function. We also evaluate two energy functions described in chapter 2, AMW and Rosetta. In what follows we first define the problem addressed in this chapter, summarize the tree-based search framework, place it context of other optimization algorithms developed for *de novo* structure prediction, and then relate our novel work on comparing various algorithmic realizations of this framework. Representative results follow. The images used in this chapter are copyright of the IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) journal.

4.1 Background and Related Work on *de novo* Protein Structure Prediction

In *de novo* structure prediction, one is provided a sequence of amino acids for a target protein, and the goal is to produce a complete specification of all atoms in terms of their cartesian coordinates in the native structure of the target protein. *De novo* structure prediction typically proceeds in two stages [103]. In the first stage, a set or ensemble of low-energy conformations are obtained. These conformations are referred to as *decoys*, as

only a subset of them may reside in the global minimum that represents the native state of the target protein sequence; that is, the decoys represent various local minima sampled in the first stage. It is the goal of the second stage to analyze the decoys and identify the native state. The first stage is referred to as decoy sampling, whereas the second as decoy selection. The object of our investigation here is the decoy sampling stage, as adequate coverage of the local minima in the energy surface of a target protein sequence is important so as not to miss the native state sought to be identified in the decoy selection stage. In the following, we provide a summary of the sampling techniques employed for decoy sampling, focusing primarily on their sampling capability. While decoy selection is also an active research area, standard clustering algorithms perform well, as long as the native state is sufficiently sampled in the first stage. It is worth noting that the identification of this state in the second stage does not rely on simply selecting the lowest-energy decoys, as it is now recognized that often the global minimum is not the native state; typically, the deepest minima are artifacts of an energy function [104]; instead, the second stage relies on clustering based on structural similarity, and reports the most-populated cluster as the native state; the intuition behind this approach is that a wider minimum is more likely to contain the native state and not be an artifact of a given energy function.

4.1.1 Predominant Stochastic Optimization Frameworks for Decoy Sampling: Molecular Dynamics versus Monte Carlo

In decoy sampling, any search technique can be used to populate local minima of the energy surface of a target protein. The two most common templates used are Molecular Dynamics (MD) and Monte Carlo (MC) search. A detailed review of these templates and their more powerful adaptations for decoy sampling can be found in [105]. In summary, when MD search is used, the search is initialized with a random or extended conformation. A series of conformations is then produced, as the search effectively follows the negative of the gradient of the energy function employed; the gradient needs to be re-evaluated often, which increases the computational demands of MD searches. Moreover, one MD trajectory leads to a local

minimum, and so, typically, many of them are launched from various initial conformations in a random restart fashion. MC searches have been shown to have higher sampling capability than MD searches, as they do not have to follow the laws of motions of the particles/atoms that make up the target protein. Instead, a conformation is generated from a given one through the usage of moves, which may be changes to selected dihedral angles or other more effective moves. This process is repeated to obtain a series of conformations. Effectively, an MC search hops in the energy surface, overall leading to low-energy conformations, while allowing high-energy moves per a probabilistic criterion referred to as the Metropolis criterion [106]. An MC search will also lead to a local minimum, so many are typically launched through random restart.

Coarse Graining and Molecular Fragment Replacement

One of the strategies employed to further enhance the sampling capability of MC-based frameworks for decoy sampling is to have them operate on coarse-grained/low-resolution representations of the amino-acid chain of the target protein. Typically, only the backbone atoms are modeled, and so the generated decoys lack side chains. Energy functions exist to score such coarse-grained decoys. It is in the second stage that, prior to clustering, each decoy is added side chains with side-chain packing techniques, thus represented at high resolution. Another important strategy, which is often credited with the greatest advancement in *de novo* structure prediction, is the employment of special moves in an MC search. These moves are referred to as fragment replacements, and the idea is to effectively discretize the search space. Instead of assigning random values to selected dihedral angles in order to generate a new conformation from a given one, a fragment of consecutive backbone dihedral angles are selected for modification. The values of all these angles are simultaneously replaced with values found for the corresponding sequence of amino acids of the fragment in known native protein structures. The pre-compilation of a fragment library is important, but a detailed description is not the focus of our work here. We employ here the Rosetta fragment libraries [107], which we have shown to effectively mine protein

structure databases [46, 108]. The impact of fragment lengths has also been investigated by researchers [109]. Typically, fragment lengths 9 and 3 are used, to alternate between sampling physically-realistic conformations fast (with longer fragments) and then searching in their vicinity for lower-energy ones (with shorter fragments).

The fragment replacements are effectively good moves that narrow the navigation of the conformation space. Given a protein conformation C_i, an amino acid t is selected. We then define a fragment of length f from amino acid t to $t + f - 1$. The amino-acid sequence of that fragment is used to query the library, and among all different configurations (sets of values for the dihedral angles of that fragment), a configuration is sampled uniformly at random to replace the one in C_i and thus yield conformation C_{i+1} in a growing MC trajectory. The replacement is accepted with probability $e^{\alpha \cdot -(E_{i+1}-E_i)}$, known as the Metropolis criterion; α is a parameter related to the notion of temperature, which controls the increase in energy accepted between two consecutive conformations. The process is repeated, either systematically, selecting amino acid $t + 1$, or at random to grow the MC trajectory.

The molecular fragment replacement technique is often credited with the greatest advancements in *de novo* structure prediction, and is now the component shared by state-of-the-art protocols, such as Rosetta [110] and Quark [111]. However, in all such protocols, the decoy sampling stage relies on random restart in order to obtain a broad view of the local minima in the energy surface. This approach does not make effective use of computational resources, as the MC trajectories are bound to lead to same or nearby minima, as they do not exchange information with one another on what regions of the conformational space and the energy surface have already been explored. In order to address this, work in the Shehu lab has proposed a different search framework that builds over MC but instead integrates the MC trajectories in a tree search structure that adaptively grows in conformational space.

4.1.2 Robotics-inspired Tree-Based Stochastic Optimization Framework

In contrast to random restart, a robotics-inspired framework, FeLTr, has been introduced in [5, 45] to effectively allow exchange of information among MC trajectories and guide the

search to both under-explored regions of the conformational space and low-energy regions of the energy surface. The tree is rooted with the extended conformation and grown through a cycle of selection and expansion operations. The expansion step consists of performing a short MC trajectory, employing the Metropolis criterion. The end point is appended to the tree as a child node (the parent node is the conformation from which the trajectory began). The selection operator controls the growth of the tree, selecting a node in the tree from which to continue the exploration. Discretization layers (and probability functions designed over them) are used to aid in the selection of a conformation residing in a low-energy region of the conformational space and in an under-explored region of the conformational space.

FeLTr is inspired from motion planning algorithms in robotics, which employ subdivisions of the robot workspace or configuration space to guide the search towards under-explored regions [112–116]. Similarly, in FeLTr, the search is adaptively guided to low-energy yet geometrically-distinct conformations through the use of two discretization layers that facilitate analysis of the explored conformational space and energy surface. The first layer is over the empirical energies of the decoy ensemble and the second is over their geometries. A 1d grid is associated with energies of conformations in the tree. For each grid cell in the energy discretization, a 3-d grid is created based on a subset of the coordinates calculated by the Ultrafast Shape Recognition (USR) algorithm [117]. This algorithm builds a feature vector for each decoy conformation based on a a set of geometric features (average distance from the centroid, average distance from the point farther from the centroid, etc). FeLTr is illustrated in Figure 4.1 and shown in pseudo-code in Algorithm 4.1.2.

Probability distribution functions can be defined over the discretization layers to bias the growth of the tree. FeLTr has been shown to have higher sampling capability than a long MC trajectory, and the combination of both discretization layers has been shown to improve sampling over using one of them in isolation or none at all (when both layers are turned off, the tree degenerates to an MC trajectory) [45]. Fragments of length 3 and the AMW energy function have been employed in previous work. On many proteins, the exploration has been found to approach the native structure within 5Å [45–47].

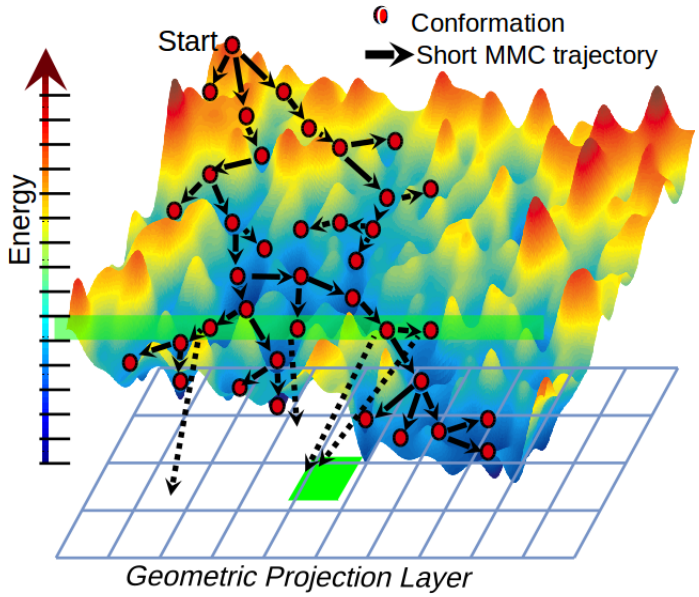


Figure 4.1: The conformation tree grown by FeLTr [5]. The conformational space is first discretized by energy (scale shown on the left), and then by geometry (projection layer shown at the bottom). A probability distribution is associated with each of the layers (which controls the growth of the search tree) dictates from which cell a conformation is selected for expansion. Each of the paths within this tree is an MC trajectory.

Algorithm 4.1 Pseudo-code for Shehu decoy sample generation framework [45]

Input:
α, amino-acid sequence
Output:
Ω, an ensemble of decoy conformations

1: $C_{init} \leftarrow$ extended coarse-grained conformation for α
2: AddConf(C_{init}, $Layer_E$, $Layer_{proj}$)
3: **while** Time_Remaining and $|\Omega| < Limit$ **do**
4: $\ell \leftarrow$ SelectEnergyLayer($Layer_E$)
5: cell $\leftarrow$ SelectGeomCell ($\ell.Layer_{proj}.cells$)
6: C $\leftarrow$ SelectConf(cells.confs)
7: $C_{new} \leftarrow$ ExpandConf(C);
8: AddConf(C_{new}, $Layer_E$, $Layer_{proj}$)
9: $\Omega \leftarrow \Omega \cup \{C_{new}\}$
10: **end while**

It is worth noting that FeLTR is a versatile framework, allowing for different algorithmic realizations to be investigated. For instance, we have explored different geometric projection layers for their ability to direct the search to under-sampled regions of conformational space [47, 118]. This work employs this versatility to investigate the role of energy in directing the search. We investigate the role of energy bias by investigating the impact of various probability distribution functions defined over the energy projection layer. We complete our treatment by following up the various algorithmic realizations of FeLTr with the clustering and a decoy selection stage in order to present blind predictions, as in the *de novo* structure prediction setting.

4.2 Methods

We briefly review in some more detail the selection mechanism in FeLTr in order to setup the various algorithmic realizations we investigate here. Next, we introduce a method for analyzing the resulting ensemble of decoys and selecting a subset of decoys for high-resolution refinement. Finally, we present a method to analyze the results of the high-resolution refinements for selecting a final set of candidate predicted structures/solutions.

4.2.1 Biasing the Exploration

Prior work on FeLTr has focused on the rapid identification of the lowest-energy conformations. To facilitate this goal, the energy grid is constructed with cells 2 kcal/mol wide. Each cell is assigned a weight via the function $w(\ell) = E_{\text{avg}}(\ell) \cdot E_{\text{avg}}(\ell) + \epsilon$, where ϵ is a small value that ensures high-energy conformations have a nonzero probability of selection. A level ℓ is selected with probability $w(\ell)/\sum_{\ell' \in \text{Layer}_E} w(\ell')$. We will refer to this probability distribution as the `QUAD` distribution. Once an energy level is selected, a cell belonging to it in the 3d geometric projection grid can be selected according to another probability distribution. A second weight function, $1.0/[(1.0 + \texttt{nsel}) \cdot \texttt{nconfs}]$, is used where `nsel` records how often a cell is selected, and `nconfs` is the number of conformations projected to the cell.

This function avoids cells that have been selected for expansion many times before and are already populated by many conformations. Once a cell is selected, any conformation in it can be selected at random for expansion; a short MMC trajectory from that conformation constitutes a new branch of the tree.

The objective in previous work has been to demonstrate that FeLTr improves coverage of the conformational space over independently-running MC trajectories. While QUAD biases the tree towards lower energies, employment of QUAD for the purpose of decoy generation risks exploiting minima that are artifacts of the energy function. However, the employment of probability distribution functions to ultimately control the distribution of sampled conformations make FeLTr particularly versatile for the purpose of decoy sampling and the study of deficiencies in *de novo* modeling. Here we propose different probability distribution functions to implement the energy bias and show that one of them, corresponding to a soft energy bias, is better suited to obtain a broad non-redundant view of the energy surface through low-energy distinct decoys. We do so on two different state-of-the-art low-resolution energy functions and show that, while both allow capturing near-native conformations in the decoy ensemble, both are capable of associating very low scores with non-native decoys. We now detail the implementation of the energy bias.

Implementing Energy Bias

The `QUAD` probability distribution function defined over weights $w(\ell) = E_{\text{avg}}(\ell) \cdot E_{\text{avg}}(\ell) + \epsilon$ described above essentially implements a strong energy bias that controls the growth of the tree through the expansion of lowest-energy decoys to obtain even lower-energy decoys. Note that the geometric projection grid is employed as above in conjunction with the energy bias. This setting can be very greedy and lead FeLTr, despite the bias away from over-sampled cells in the conformational space, to deep energy minima that are artifacts of a given energy function. In contrast, one can ignore energy bias altogether. Essentially, all conformations can be treated as energetically equivalent and projected to the same energy level. Only the geometric projection grid and the probability distribution function defined

on it (defined above over weights $1.0/[(1.0 + \texttt{nsel}) \cdot \texttt{nconfs}]$) can be employed. Let us refer to this probability distribution function as `COV`, as it essentially allows ignoring the energy surface and only steers the search to coverage of unsampled regions of the conformational space.

A new probability distribution function can be defined to implement a soft energy bias instead. As the tree and its conformational ensemble Ω grow, the mean (μ_Ω) and standard deviation (σ_Ω) can be updated over the energies of decoys. The mean tends to go lower over time, as the MMC trajectories that constitute the tree branches guide the tree towards lower energies through the Metropolis criterion. The energy level whose average energy is closest to a sample drawn from the Gaussian distribution $(\mu_\Omega, \sigma_\Omega)$ can be selected for expansions. The geometric projection grid is employed as above. We refer to this third realization of the framework as `NORM`. Unlike `QUAD`, `NORM` does not greedily bias the search tree towards the lowest-energy decoys. Instead, the tree slowly grows towards low-energy decoys and associates low probabilities of selection to energy levels on either tail of the energy distribution.

4.2.2 Employed Representation and Energy Functions

We recall that, when employing the AMW energy function, the representation reduces side chains to only the C_β atom (with exception of glycine). When employing the Rosetta energy function, the C_β atom is swapped for a centroid per side chain. Internally, two representations are maintained, one angular and another consisting of cartesian coordinates. The angular representation maintains only three backbone dihedral angles (ϕ, ψ, ω) per amino acid, as sampled from the fragment configuration library.

4.2.3 Ensemble Analysis

We now describe techniques to compare the different realizations of FeLTr implementing the three different energy biases described above.

Energetic Reduction

Reducing the decoy ensemble Ω produced by the tree through an energetic criterion allows removing high-energy decoys added to the tree during the exploration. We employ a non-parametric threshold that discards any sampled conformation with energy higher than the mean. This threshold is not protein-dependent and reduces the size of the ensemble by about 50%. While discarding about half the ensemble may sacrifice a few decoys with low lRMSDs to the native structure, the majority of low-lRMSD decoys are generally maintained in the reduced ensemble Ω_E. The results in section 4.3 show that more low-lRMSD conformations are maintained when reducing the ensemble produced through `QUAD` and `NORM`. This is expected, as these two probability distribution functions implement an energy bias, and near-native conformations, while not among the lowest energy decoys, are associated with low energies. The results in section 4.3 also show that more near-native conformations are retained when reducing the ensemble produced through `NORM` than `QUAD`, and this is particularly pronounced when using the AMW versus the Rosetta energy function.

Geometric Reduction

FeLTr employs coarse projection coordinates to efficiently group together similar conformations and bias the search on the fly away from oversampled regions. Employing lRMSD-based comparisons and clustering would provide more detail and accuracy, but it would not be efficient. However, lRMSD-based clustering can be performed on the energetically-reduced ensemble Ω_E both to analyze and compare the diversity of decoys across the three realizations of FeLTr and to further reduce the ensemble to a subset of distinct regions from which exploration can resume at greater detail.

We utilize an adaption of the bisecting K-Means algorithm [119] on the Ω_E ensemble. Medoids instead of centroids are chosen to represent clusters so as to avoid irregular local structures resulting from angle averaging [120]. Initially, a conformation is selected at random to serve as the representative of the first cluster that encompasses all conformations in the ensemble. The essential process in bisecting K-Means clustering is that a cluster is

broken into two new ones if the minimum lRMSD from their cluster representative is above an ϵ threshold. Two random conformations are selected to serve as the representatives of the two new clusters. When conformations are reassigned, the representatives selected at random are replaced with the cluster medoids. The proximity of the conformations in each cluster is reevaluated. If the minimum lRMSD is above ϵ, the process begins anew (hence, bisecting). In the end, the medoids of the clusters are essentially a reduced representation of the Ω_E ensemble and constitute the $\Omega_{E,C}$ ensemble.

The bisecting K-Means algorithm is less susceptible to initialization issues and does not require a priori determining the number of clusters. It requires, however, setting the maximum intra-cluster distance ϵ. In this work, we analyze the effect of two different values, 3 and 5Å on the diversity of the resulting $\Omega_{E,C}$ ensemble.

4.2.4 Exploration Convergence

The reduced ensemble $\Omega_{E,C}$ can now be used to drive the exploration towards possible convergence on a more complex search space. A long MMC trajectory is launched from each conformation in $\Omega_{E,C}$. The trajectory length is a compromise between reaching convergence and controlling the overall computational cost. The fragment length employed here is 3 (9 is used by the frameworks above to obtain Ω). The shorter fragment length increases the complexity of the conformational space but also allows adding more detail to the energy surface.

The end points of the trajectories are analyzed through density-based clustering analysis [120]. An end point is assigned the number of neighbors that are within an lRMSD threshold of it (we use the same ϵ threshold above). The end point with the largest number of neighbors is considered to be the representative of the most populous cluster. This point and its neighbors are removed, and the process continues until all conformations have been exhausted. An exploration that started with obtaining a broad view of the energy surfaces terminates with revealing decoys in regions of the conformational space where many MMC trajectories converge. The results in section 4.3 show that near-native conformations are

retained among the top populous clusters; that is, the corresponding decoys are near-native and as such are good candidates for high-resolution refinement.

4.3 Results

To test the effectiveness of the proposed energy biases, we test on ten different protein systems, listed in Table 4.1. The systems range from 61-123 amino acids in length, cover α, β, and α/β folds, and include CASP targets. The list includes sequences longer than 70 amino acids and α/β native topologies known to be challenging for *de novo* structure prediction.

Table 4.1: The PDB ID, nr. of amino acids, and known native topology are shown for the 10 proteins studied.

ID	1gb1	1sap	1wapa	1fwp	1ail	1aoy	1cc5	2ezk	3gwl	2h5nD
N	56	66	68	69	70	78	83	93	106	123
Fold	α/β	α/β	β	α/β	α	α/β	α	α	α	α

The main measurement used in the analysis below is lRMSD (discussed in section 2.3.1). Each biasing scheme using each of the two energy functions is applied on each protein for 24 CPU hours on a 2.66 GHz Opteron processor with 8 GB of memory. This is repeated three times to obtain 3 ensembles per setting. Results and further analysis are presented on the ensemble that yields the median value in terms of lowest lRMSD from the native structure (lRMSD is calculated over heavy backbone atoms). Clustering is conducted on a 2.4 Intel Xeon E5620 processor with 24 GB of memory. The MMC trajectories that optimize each decoy in the resulting ensemble $\Omega_{E,C}$ are limited to $20,000$ steps and are run on a 2.66 GHz Opteron processor with 8 GB of memory. This second stage lends itself to embarrassing parallelization and takes 12-36 hours on 80 CPU cores depending on the size of $\Omega_{E,C}$ and protein length.

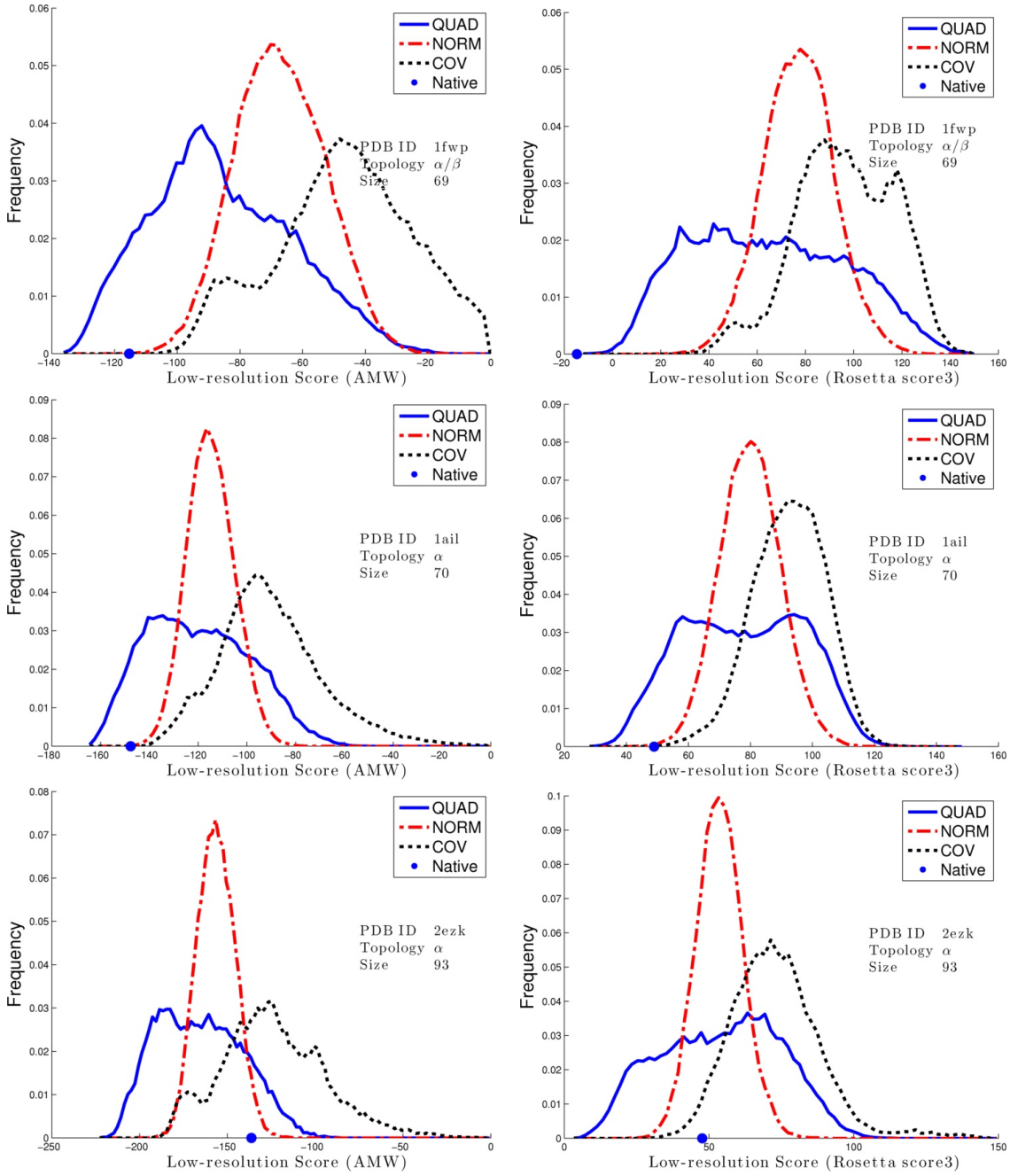


Figure 4.2: Distributions of energies of Ω resulting from `QUAD`, `COV`, and `NORM` are superimposed over one another. The energy of the native structure is marked by a blue circle on the x-axis. While the top row shows results obtained with AMW, the bottom row shows results obtained with the Rosetta score3 function.

Table 4.2: The lowest lRMSD from the native structure is shown for each of the three biasing schemes. Results are shown for both AMW and Rosetta score3.

	lowest lRMSD(Å) over Ω					
	AMW			Rosetta score3		
ID	COV	QUAD	NORM	COV	QUAD	NORM
1gb1	4.7	5.0	4.6	4.4	**3.8**	4.1
1sap	6.8	6.5	5.2	5.9	5.9	**4.5**
1wapa	7.6	7.4	6.9	**6.4**	6.8	6.6
1fwp	6.6	6.9	6.1	5.8	5.1	**4.6**
1ail	3.5	2.5	**1.9**	4.7	4.7	4.6
1aoy	5.5	5.6	5.8	**5.0**	5.2	5.4
1cc5	5.9	**5.7**	5.8	6.5	5.9	5.8
2ezk	4.5	3.7	4.1	3.2	**3.1**	3.5
3gwl	6.1	5.5	6.0	**4.6**	6.0	6.5
2h5nD	9.0	**6.9**	9.0	8.9	9.9	11.1

4.3.1 Analysis of Decoy Ensembles Obtained with Different Biasing Schemes

The distribution of conformational energies in Ω is shown for QUAD, COV, and NORM in Figure 4.2 on three selected proteins. Superimposition of the distributions shows that, as expected, QUAD results in lower energies (distribution is shifted to the left), whereas COV results in higher energies. The distribution obtained with NORM is expectedly Gaussian, and its mean energy is between the means of QUAD and COV. Each of the three distributions can contain lower energies than the native structure, whose energy is shown for reference.

Figure 4.2 shows these results when either AMW or Rosetta score3 are employed. Due to detailed fine tuning in calculations of the Rosetta energy functions, the setting with Rosetta score3 runs 6-7 times faster than when employing our in-house version of AMW. In order to conduct a fair comparison, the size of the conformational ensemble obtained when using Rosetta score3 is limited to the size obtained in 24 hrs with AMW on a particular protein and biasing scheme. For instance, if within 24 CPU hours, the ensemble obtained with AMW on the system with PDB ID 1fwp is 51K when using QUAD and 95K when using NORM, the ensemble sampled when using Rosetta score3 and QUAD is then limited to 51K conformations, and the ensemble sampled when using Rosetta score3 and NORM is limited

to 95K conformations.

It is worth noting that one cannot directly compare values between the AMW and Rosetta energy functions. However, the location of the known native structure shows that both energy functions can associate low or high energies with a native structure. For instance, on the protein systems with PDB IDs 1fwp and 1ail, the native structure has lower energy than the mean of the energy distribution obtained under `NORM` whether AMW or Rosetta score3 are employed. On the system with PDB ID 2ezk, the native structure has higher energy than the mean under AMW but not Rosetta score3. On all three systems, lower energies than that of the native structure can be obtained under `QUAD` under each energy function due to the strong energy bias in `QUAD` driving the exploration towards deep non-native minima.

Table 4.2 shows the lowest lRMSD obtained under each biasing scheme when using AMW or Rosetta. As in Figure 4.2, the data are presented on the median ensemble (over three runs for each biasing scheme). Lowest lRMSDs under 6Å are obtained by all three biasing schemes on most protein systems, whether AMW or Rosetta score3 are used. The global energy bias present in `QUAD` and `NORM` but not in `COV`, improves proximity to the native structure (lower minimum lRMSDs are obtained overall). Moreover, when using AMW, lower minimum lRMSDs are obtained on 50% of the systems with `NORM` than `QUAD`, comparable lowest lRMSDs within 0.2Å are obtained on 20% of the systems, and increases are observed on the rest. When using the Rosetta energy function, differences in lowest lRMSDs between `NORM` and `QUAD` are less pronounced, suggesting than the Rosetta energy surface is more complex than AMW and can benefit from further sampling. A comparison between AMW and Rosetta score3 reveals that the lowest lRMSD is obtained by Rosetta score3 (in bold) for most systems, whether `COV`, `NORM`, or `QUAD` are used. AMW seems to have a significant advantage on 1ail and obtains comparable results on 1cc5, both all-α proteins. Results are uniformly poor on 2h5nD, suggesting that this large protein may benefit from further sampling.

Focusing on the lowest lRMSD may be misleading, as the conformation realizing it

may not be sufficiently represented in the decoy ensemble or may be missed altogether by a selection technique. Figure 4.3 analyzes Ω in more detail for 3 selected protein systems. The 20 decoys with the lowest lRMSDs from the native structure are marked in the distribution of conformational energies obtained with each biasing scheme.

Figure 4.3 shows that many of the 20 lowest-lRMSD conformations can be lost if the selection criterion discards those with energies above the mean in the ensembles obtained with AMW and `QUAD`. Many of these conformations would be retained if using `NORM`. Differences between `QUAD` and `NORM` are less pronounced when using Rosetta, suggesting again that the Rosetta energy surface is more complex. We point out that the system with PDB ID 1ail, an all α protein, seems to be an easier case for AMW than Rosetta. Whether using `QUAD` or `NORM` with AMW, the 20 lowest-lRMSD conformations have energies not only below the mean but also close to that of the native structure. On the other hand, the system with PDB ID 2ezk seems to be more challenging for AMW than Rosetta. When using AMW, the 20 lowest-lRMSD conformations have energies that place them above the mean whether using `QUAD` or `NORM`. In contrast, when using Rosetta score3, many of these conformations are close in energy to the native structure, which also falls below the mean both under `NORM` and `QUAD`. We note that this system is a longer α protein of 93 amino acids.

A further comparisons between AMW and the Rosetta energy function can be conducted by comparing not only the lowest lRMSDs or the highest GDT_TS scores to the known native structure obtained on each system but also the mean lRMSD and the mean GDT_TS score on the 90% percentile of low-energy conformations. The results shown in Table 4.3 fix the biasing scheme to `NORM` and limit the source of variation to the energy function employed. Values in bold indicate either lower or comparable lRMSDs between AMW and Rosetta or higher or comparable GDT_TS scores between AMW and Rosetta. If focusing on lowest lRMSDs, Rosetta provides scores that are lower or comparable than those obtained with AMW on 7/10 of the systems. Looking at GDT_TS scores brings the number of systems with higher or comparable GDT_TS scores in Rosetta to 8/10. Interestingly, the majority of the improvements are on proteins with β or α/β folds. On the majority of the all-α

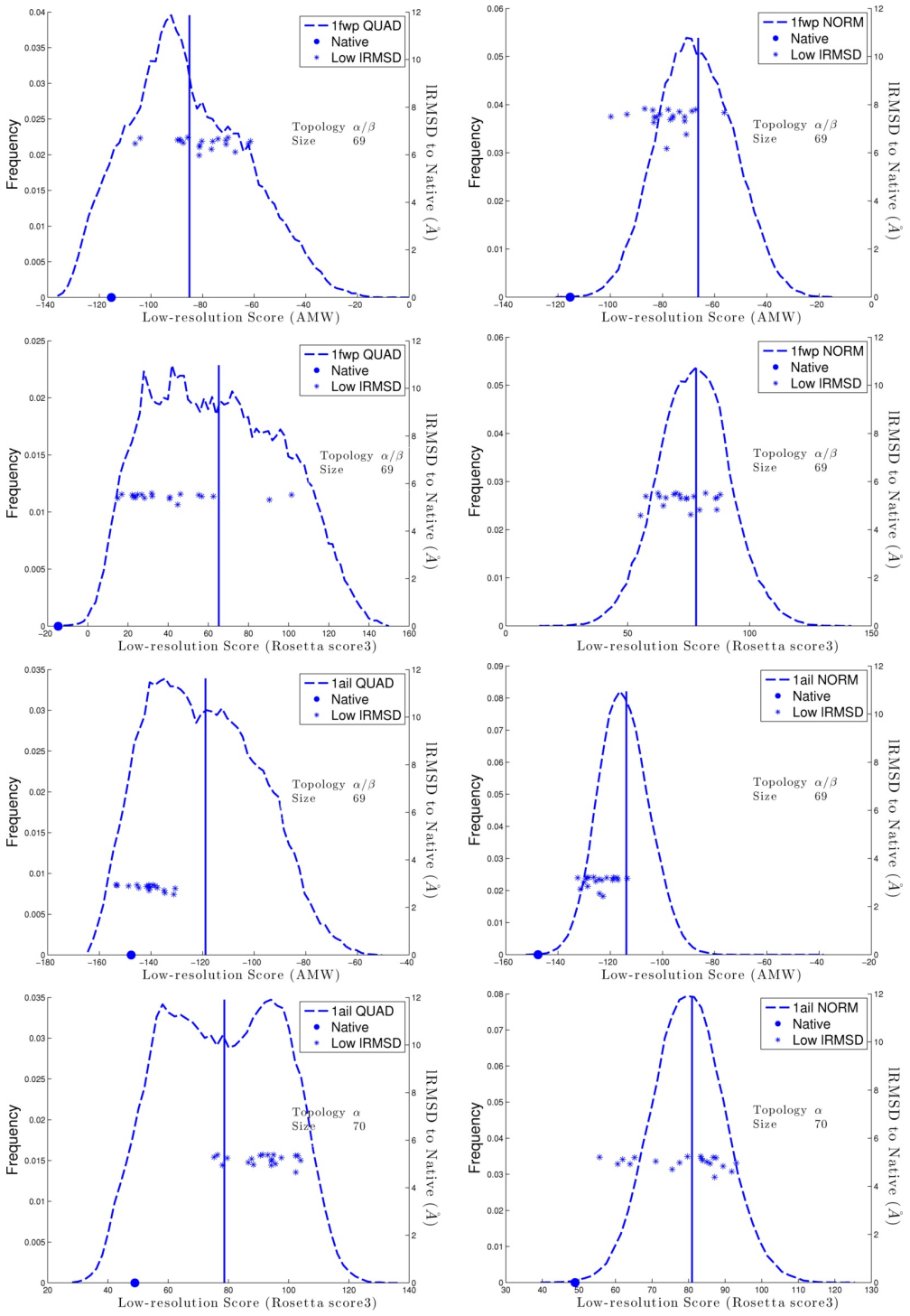


Figure 4.3: The 20 lowest-lRMSD conformations are shown as blue circles over the distribution of energies in Ω for 2 selected protein system. Their lRMSDs from the native structure are shown on the right hand axis. Results are shown for both AMW and Rosetta score3.

proteins, AMW provides better or similar results.

Comparing mean lRMSDs and mean GDT_TS scores over the 90th percentile of low-energy conformations reveals that differences between Rosetta and AMW in terms of representation of near-native conformations are less stark. Rosetta has lower or comparable mean lRMSDs or higher or comparable mean GDT_TS scores on this subensemble of conformations on 30% and 70% of the systems, respectively. Taken together, these results provide a detailed insight into AMW and Rosetta. While Rosetta seems capable of better recognition of conformations in close proximity to the native structure, neither energy function has a distinct advantage for the purpose of a selection technique driven by an energy cutoff.

Table 4.3: AMW and Rosetta energy functions are compared over entire Ω ensemble obtained with `NORM`. In addition to lowest lRMSD and maximum GDT_TS to the known native structure, the comparison includes mean lRMSD and mean GDT_TS over the 90th percentile (p90) of low-energy conformations in Ω.

	$\text{lRMSD}_{\min}$(Å)		$\text{GDT\_TS}_{\max}$(%)		$\text{lRMSD}_{\mu,p90}$(Å)		$\text{GDT\_TS}_{\mu,p90}$(%)	
ID	AMW	Rosetta (score3)	AMW	Rosetta (score3)	AMW	Rosetta (score3)	AMW	Rosetta (score3)
1gb1 (α/β)	4.6	**4.1**	0.63	**0.69**	11.4	**9.3**	0.39	**0.49**
1sap (α/β)	5.2	**4.5**	**0.52**	**0.52**	**10.6**	11.9	**0.34**	**0.32**
1wapa (β)	6.9	**6.6**	0.39	**0.43**	**13.0**	13.7	0.23	**0.28**
1fwp (α/β)	6.1	**4.6**	0.48	**0.53**	12.4	**11.1**	0.30	**0.35**
1ail (α)	**1.9**	4.6	**0.84**	0.65	**9.8**	11.0	**0.43**	0.37
1aoy (α/β)	5.8	**5.4**	0.57	**0.62**	**9.9**	12.2	**0.40**	0.36
1cc5 (α)	**5.8**	**5.8**	**0.45**	**0.46**	**12.3**	13.2	0.28	**0.30**
2ezk (α)	4.1	**3.5**	0.56	**0.70**	11.7	**8.4**	0.34	**0.50**
3gwl (α)	**6.0**	6.5	**0.44**	**0.46**	**13.4**	15.6	**0.30**	**0.31**
2h5nD (α)	**9.0**	11.1	**0.33**	0.24	**15.5**	16.7	**0.23**	0.19

4.3.2 Ensemble Reduction and Analysis

Since our goal for the robotics-inspired exploration is to obtain a broad non-redundant view of the energy surface, `QUAD` and `NORM` are further investigated in terms of the geometric diversity of the Ω_E ensembles they yield (discarding any conformation with energy above

the mean). Since the bisecting K-Means clustering employed for this purpose makes use of an $N \times N$ matrix to store pairwise lRMSDs between the N decoys in Ω_E, the size of Ω_E can pose computational and memory issues. We impose a limit of 40K conformations. When the limit is exceeded, uniform sampling over Ω_E is used to obtain 40K conformations. Table 4.4 shows $|\Omega|$ and $|\Omega_E|$ for each protein in columns 2-3 for `QUAD` and 6-7 for `NORM`. Larger Ω ensembles are obtained on all proteins with `NORM`, confirming that it becomes increasingly harder to satisfy the Metropolis criterion (and so expand selected conformations) from the lowest-energy levels selected by `QUAD`. The difference in $|\Omega|$ between `QUAD` and `NORM` becomes less pronounced on the longer proteins, where energy evaluations become the bottleneck.

Table 4.4: $|\Omega|$ and $|\Omega_E|$ obtained when using AMW are shown in units of 10^3. Δ_C shows $|\Omega_E| - |\Omega_{E,C}|$ as a % of Ω_E. Subscripts 3 and 5 refer to ϵ values 3 and 5Å .

ID	AMW							
	`QUAD`				`NORM`			
	$\|\Omega\|$	$\|\Omega_E\|$	ΔC_3	ΔC_5	$\|\Omega\|$	$\|\Omega_E\|$	ΔC_3	ΔC_5
1gb1	101	40	57%	83%	168	40	28%	65%
1sap	70	40	76%	90%	105	40	35%	51%
1wapa	45	26	78%	86%	84	42	37%	52%
1fwp	51	33	73%	88%	95	40	31%	51%
1ail	73	38	76%	90%	94	40	58%	80%
1aoy	57	31	73%	90%	71	35	47%	72%
1cc5	37	33	71%	83%	55	28	32%	43%
2ezk	38	20	63%	87%	42	21	43%	85%
3gwl	23	12	70%	85%	28	14	47%	75%
2h5nd	15	8	61%	76%	18	9	55%	69%

The reduction in size of $\Omega_{E,C}$ resulting from the clustering of Ω_E is shown in columns 4-5 and 8-9 of Table 4.4 for `QUAD` and `NORM`. Results are shown for ϵ values of 3 and 5Å (a higher value would degenerate the quality of the clusters). As expected, a higher ϵ value results in a more significant reduction over Ω_E. Moreover, comparison between `QUAD` and `NORM` for a given ϵ shows that clustering is able to achieve a more substantial reduction on the Ω_E

ensemble resulting from QUAD. This suggests that NORM results in a more diverse set of low-energy decoys, and so it is better suited to be employed for the purpose of obtaining a broad view of the energy surface. The improved diversity of low-energy decoys implies increased coverage of the conformational space, which is a critical component, especially if it is to be followed by further more detailed exploration or studies focusing on improvements of energy functions on a diverse set of decoys. The results shown in Table 4.4 are overall reproduced when using Rosetta score3, shown in Table 4.5. A more substantial reduction is obtained on the ensemble obtained with QUAD using Rosetta score3, as well, further suggesting that the soft energy bias in NORM is more appropriate at yielding a diverse non-redundant decoy ensemble not exploiting artifacts of an energy function.

Table 4.5: $|\Omega|$ and $|\Omega_E|$ obtained when using Rosetta score3 are shown in units of 10^3. Δ_C shows $|\Omega_E| - |\Omega_{E,C}|$ as a % of Ω_E. Subscripts 3 and 5 refer to ϵ values 3 and 5Å .

ID	Rosetta score3							
	QUAD				NORM			
	$\|\Omega\|$	$\|\Omega_E\|$	ΔC_3	ΔC_5	$\|\Omega\|$	$\|\Omega_E\|$	ΔC_3	ΔC_5
1gb1	101	50	70%	92%	168	40	65%	90%
1sap	70	33	69%	84%	105	52	54%	76%
1wapa	45	25	85%	95%	84	41	38%	65%
1fwp	51	26	70%	84%	95	47	50%	75%
1ail	73	36	71%	85%	94	47	57%	78%
1aoy	57	29	70%	89%	71	36	49%	83%
1cc5	37	18	80%	87%	55	28	64%	77%
2ezk	38	18	70%	93%	42	21	64%	94%
3gwl	23	12	72%	91%	28	14	48%	67%
2h5nd	15	8	76%	88%	18	9	52%	78%

4.3.3 Convergence Analysis

Here we conduct further analysis and optimization of obtained decoys. The conformations in Ω_{E_C} (medoids of clusters) resulting from NORM now serve as starting points for MMC

trajectories ($20,000$ steps long). Unlike the previous stage, which uses fragments of length 9, the MMC trajectories use fragments of length 3. The end points of the trajectories constitute the final set of conformations subjected to density-based analysis to detect possible regions of convergence.

The quality of the top 10 clusters resulting from the density-based analysis with ϵ=5 is shown for each of the protein systems in Table 4.6. The results shown in Table 4.6 are obtained with AMW. Columns 2-4 show the lowest lRMSD from the native structure over the representatives of the top i populous clusters, where i varies from 10, 5, down to 1, respectively. For reference, columns 5-6 show the tenth lowest lRMSD and the lowest lRMSD over the entire $\Omega_{E,C}$ ensemble. Additionally, columns 7-8 show the lRMSD of the conformation that can be assembled if the fragment configuration selected from the library for each fragment is the one that is closest to the actual fragment configuration in the native structure (a process known as global fit [45]).

Comparison of these columns allows drawing a few conclusions. If either the top 5 or top 10 populous clusters are employed for further refinement, near-native decoys (in terms of low lRMSDs) are preserved after the selection, promising recovery of the native structure in great detail and accuracy. Comparison of columns 4 and 5 shows that at most the selection loses $\approx$ 4Å in terms of proximity to the native structure and on average loses 1.5Å. In general, there is good correlation between cases when low lRMSDs are maintained by the selection and low lRMSDs obtained by global fit. Lower lRMSDs obtained over global fit suggest that sometimes suboptimal fragment configurations are needed locally in order to obtain a better global conformation. Similar observations can be drawn from the density analysis over ensembles obtained with Rosetta score3. The Rosetta score3 improves the quality of the lowest lRMSD among the top ten clusters on some systems but it offers no distinct advantage overall (data not shown).

Further detailed analysis is showcased on 3 representative systems. The density-based analysis is repeated on the set of conformations resulting after every $2,000$ MMC steps (AMW is used) and the aggregate size of the top i populous clusters $i \in \{1, 5, 10\}$ is shown

Table 4.6: The lowest lRMSD from the native structure over conformations in the top i clusters ($i \in 1, 5, 10$) are shown in columns 2-4, respectively. The tenth lowest and the lowest lRMSD over the entire $\Omega_{E,C}$ are shown for reference in columns 5-6, respectively. The lRMSD of the conformation resulting from global fit with fragment lengths of 9 and 3 are shown in columns 7-8, respectively.

ID	lRMSD to Native (Å)						
	T_1	T_5	T_{10}	B_{10}	B_1	G_{f9}	G_{f3}
1gb1	11.2	11.2	10.7	6.6	6.1	3.7	9.0
1sap	6.4	6.4	6.4	6.8	5.7	8.4	6.4
1wapa	10.4	10.4	9.0	7.5	6.1	17.8	6.3
1fwp	11.9	9.5	9.5	6.7	5.9	11.0	17.0
1ail	7.2	4.1	4.1	3.9	3.4	2.1	1.5
1aoy	7.1	7.1	6.9	6.0	5.0	12.9	11.5
1cc5	8.9	8.9	8.2	6.3	5.6	6.0	5.6
2ezk	7.9	7.4	7.4	5.9	4.8	10.4	9.8
3gwl	9.1	6.8	6.5	6.3	5.5	16.2	10.7
2h5nd	12.0	11.4	11.4	9.4	8.4	7.8	8.0

in Figure 4.4(a)-(c) for each system. The results in (a)-(c) showcase that this aggregate size can decrease, settle, or grow. A decrease is the result of MMC trajectories diverging in the energy surface. In (b), which shows results for the system with PDB ID 1ail, the most populated clusters grow in size, signaling convergence of many MMC trajectories to nearby regions for this system; the clusters contain a large percentage of the decoys when ϵ=5Å. Repeating the analysis with ϵ=3Å shows that 3Å is too small to measure convergence (data not shown). Convergence on the system with PDB ID 1ail suggests that the widest low-energy basins captured with AWM and `NORM` are also deep enough for the ensuing MMC runs to remain trapped. This result provides further insight into why it is that the low-resolution exploration of the AMW energy surface for this system can capture decoys within 2Å of the native structure. In contrast, the other two systems have shallower basins in the AMW energy surface.

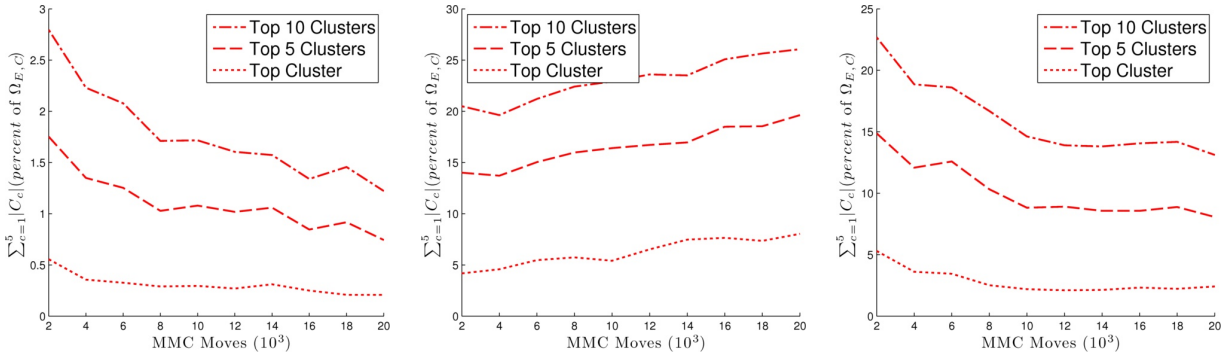
Figure 4.4(d)-(f) provides some more detail on the system with PDB ID 1ail. The distribution of energies vs. lRMSDs from the native structure of the conformations (medoids) in $\Omega_{E,C}$ in (d) shows that AMW is weakly-funneled over the 9-mer space. Figure 4.4(e)

shows that the correlation between low energies and low lRMSDs improves after the MMC trajectories populate the 3-mer space. Moreover, a proof of concept analysis takes the top 10 clusters resulting from the density-based for this system and subjects them to short high-resolution refinement through the Rosetta relaxation protocol. The resulting energetic and lRMSD ranks shown in Figure 4.4(f) make the case that the top 10 clusters are good-quality candidates for further refinement. The same analysis is repeated over ensembles obtained with Rosetta score3 on this system, shown in Figure 4.4(g)-(i). In contrast to AMW, Rosetta yields stronger funneling on the 3-mer space despite the lowest lRMSD to the native structure being higher than what is obtained with AMW. Results showing ranks after high-resolution refinements of the top 10 clusters in Figure 4.4(i) are similar to those obtained with AMW.

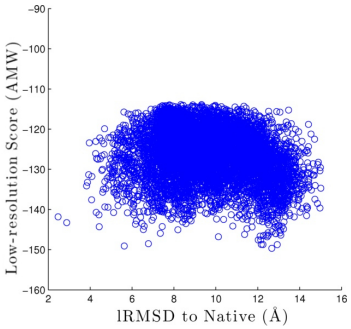
4.4 Conclusions

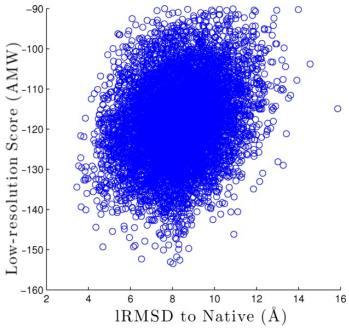
Our analysis of different probability distribution functions over the discretization layers shows that a Gaussian distribution is more suitable for a diverse ensemble of low-energy decoys. This distribution effectively implements a soft energy bias that guards the framework from converging too fast to deep energy minima. While additionally enforcing structural diversity through the geometric projection layer, the combination of a soft energy bias and coverage result in a diverse ensemble of low-energy decoys. A non-parametric energetic reduction and a K-means bisecting clustering algorithm allow further reducing the ensemble and show that near-native conformations are more likely to be retained when using the soft energy bias rather than more greedy schemes.

Comparison of ensembles obtained with AMW versus Rosetta allow drawing a few observations. First, Rosetta allows improvements in terms of closer proximity to the known native structure by as much as 1.5Å over AMW. This is more pronounced for proteins with all β or α/β folds. AMW instead is better suited for all α proteins. This observation



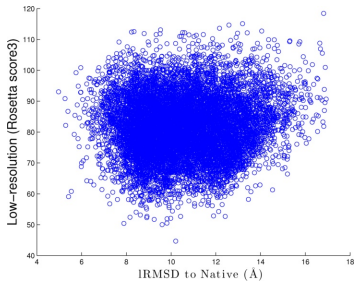
(a) 1fwp (α/β) (b) 1ail (α) (c) 2ezk (α)

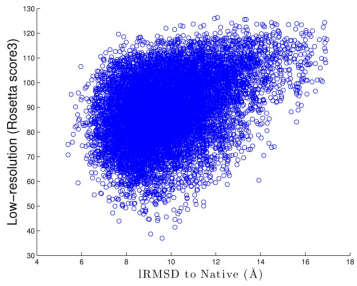




C Rank	Energy Rank	lRMSD Rank
1	7	6
2	4	8
3	8	1
4	3	4
5	9	3
6	2	10
7	1	7
8	6	2
9	10	9
10	5	5

(d) 1ail $\Omega_{E,C}$, AMW (e) 1ail after 20K MMC steps, AMW (f) High-resolution refinement





C Rank	Energy Rank	lRMSD Rank
1	5	5
2	10	9
3	6	6
4	2	4
5	4	3
6	8	7
7	3	2
8	9	8
9	7	10
10	1	1

(g) 1ail $\Omega_{E,C}$, Rosetta (score3) (h) 1ail after 20K MMC steps, Rosetta (score 3) (i) High-res. refinement

Figure 4.4: (a)-(c)The aggregate size of the top i clusters $i \in \{1, 5, 10\}$ resulting from density-based analysis with $\epsilon = 5$Å is shown every 2K MMC steps (red lines). (d)-(f) Energy vs. lRMSD from the native structure are plotted for system with PDB ID 1ail for conformations in $\Omega_{E,C}$ in (d) and for the end points of the MMC trajectories in (e). These results are obtained with AMW and `NORM`. (f) also shows the energetic and lRMSD ranking of the top 10 populous cluster representatives after a short high-resolution refinement. (g)-(i) Energy vs. lRMSD from the native structure are plotted for system with PDB ID 1ail for conformations in $\Omega_{E,C}$ in (g) and for the end points of the MMC trajectories in (h). These results are obtained with the Rosetta score 3 energy function and `NORM`. (i) also shows the energetic and lRMSD ranking of the top 10 populous cluster representatives after a short high-resolution refinement.

confirms recent analyses of versions of AMW in [49, 52, 121] that the function seems well-equipped to capture the basin of all α fold proteins. In line with other studies of the Rosetta energy function [39–41], our analysis shows, similar to AMW, energies values lower than that of the native state can be assigned to decoys with non-native topologies. A comparison of the different energy biasing schemes when using the Rosetta energy function indicates that the function results in a more complex surface than AMW. While the AMW surface is saturated more speedily by the framework, the Rosetta energy surface may benefit from further sampling.

The convergence analysis is conducted by applying long MC trajectories to the reduced ensemble. Shorter fragment lengths of 3 instead of 9 are used to access a more detailed energy surface and further populate the regions indicated as promising by the above exploration. Switching from longer to shorter fragments during exploration is employed by other methods for structure prediction [110]. These methods perform this switch in the context of very long independent MMC trajectories. In this framework, longer fragments are used to gain a broader view of conformational space. Once the areas of interest are identified via energetic reduction and geometric clustering, shorter fragments are employed to optimize the energy function on the remaining ensemble. Density-based clustering over the end points of the trajectories shows that the top populous clusters retain near-native conformations which can be used for further refinement in a blind prediction setting for *de novo* structure prediction.

Taken together, the obtained results suggest that FeLTr is versatile and allows exploring current open issues and deficiencies in *de novo* structure prediction. The density clustering analysis showcases that the enhanced sampling by the robotics-inspired framework results in many regions, including non-native topologies, being sufficiently populated to be reported among the top 10 populated clusters. This result effectively indicates that the framework leads to a diverse set of highly-populated energy basins. These basins can be used for for further development of scoring functions to improve recognition of non-native topologies.

Chapter 5: Computing Protein Motions with a Novel Tree-based Robotics-inspired Method

The work described in this chapter is based on the preliminary work published in a conference proceeding [122] and an extended version published in a journal article [1]. The problem addressed in this work is to extend our structural characterization to dynamic proteins that switch between different structural states to modulate their biological function. Specifically, we propose an efficient algorithm to compute molecular motions employed by dynamic proteins in switching between different functionally-relevant structural states. Understanding how proteins modulate their biological function at the level of structure is an important problem. It is also an important first step to elucidating at a microscopic level how perturbations, including sequence mutations, affect function. The problem is challenging for both the wet and dry laboratories. In the following we setup the problem in greater detail and provide a summary of the state of the art before proceeding to describe the framework we propose and the analysis to validate this framework. As in other chapters, we focus primarily on the methodological novelty and relate representative results. The images used in this chapter are copyright of the BMC Structural Biology Journal.

5.1 Background and Related Work on Molecular Motion Computation

Experimental evidence is now available illustrating that some protein molecules can act as molecular machines and exploit a set of thermodynamically-stable structures to vary their function [28–32]. In most cases, either no structural information exists on the conformations employed by a protein molecule to transition from one structural state to another, or this information is rather limited. One reason for the scarcity of structural information is the

inability of experimental techniques to structurally track a transition. Probing the transition at the sub-nanometer scale, as required to elucidate structures along the transition, is in principle possible with spectroscopic techniques, such as FRET or NMR. Doing so in practice is difficult, as the actual time spent during a transition event can be short compared to the long time a protein can spend in a stable or meta-stable structural state. In other words, experimental techniques are currently suitable for catching proteins in long-lived stable or meta-stable states but not in the short-lived ones that a protein uses to transition between stable and meta-stable states. There are now many stable and meta-stable states deposited for dynamic protein systems in the PDB. Examples include Calmodulin and Adenylate Kinase, which are also subjects of our investigation in this thesis.

Given the current challenges in the wet laboratory, computational methods provide an alternative approach. These techniques are in principle able to explore the protein energy surface at great detail and so compute conformational trajectories.

5.1.1 Problem Statement

The input consists of two PDB-obtained structures (start and goal) corresponding to two experimentally-determined functional states of a protein. The output is a set or ensemble of conformational paths. Each path is a series of conformations, initiated at the start structure and terminated within some threshold distance of the goal structure. The path needs to satisfy additional constraints. The energetic difference between conformations in a path is controlled via different means, either by placing a bound on the maximum energy in a path or through the Metropolis criterion. A template is illustrated in Algorithm 5.1.

5.1.2 Related Work on Molecular Motion Computation

Designing computation methods to address the above problem with reasonable computational resources remains challenging [123], as transition trajectories may span multiple length and time scales, often connecting structural states more than 100Å apart. This length scale is up to 2 orders of magnitude larger than a typical interatomic distance of

2Å. Transitions can also demand even larger μs-ms time scales, which is $6-12$ orders of magnitude larger than typical atomic oscillations of the fs-ps timescale.

These characteristics make the computation of transition trajectories exceptionally challenging for the standard random restart MD- or MC-based sampling framework. It is worth reiterating that given the stochastic nature of molecular motions, a protein system may use different pathways to access two different structural states; some of these pathways may require less work from an energetic point of view. Hence, sampling-based frameworks are needed to sample such pathways and provide a broader picture that can then identify intermediate structural states employed by a protein on most pathways to reach a goal state. It is very costly to navigate the protein energy surface in search of transition trajectories with equilibrium MD-based approaches [11, 124, 125]. A simulation may spend a long time in a local minimum corresponding to a stable or semi-stable state and only rarely undergo a conformational change allowing it to cross an energy barrier and transition to another state. Long simulation times may be needed to capture the rare event in a transition trajectory, which makes equilibrium MD-based approaches as inadequate as the NMR and FRET experimental techniques in this setting.

Algorithm 5.1 Setup of the conformational path computation problem

Input: P_{start}, P_{goal} ▷ pair of functional states

Output: $\Omega = \left\{\Pi_{ij}^{(1)}, \Pi_{ij}^{(2)}, ..., \Pi_{ij}^{(n)}\right\}$ ▷ ensemble of paths

$\forall p \in \Omega$

- $\Pi_{ij}^{(p)} = C_{ij,0}, ..., C_{ij,t}, ..., C_{ij,\tau}$
  - $0 \leq t \leq \tau$
  - $i \neq j;\ C_i = C_{ij,0};\ C_j = C_{ij,\tau}$
  - $Valid(\Pi_{ij}^{(p)}) = True$

MD-based Approaches

Many adaptations are pursued to lower the computational demands of MD-based approaches. Essentially, MD-based methods for elucidating transitions incorporate some suitable bias at the expense of obtaining possibly different transition trajectories. Methods include targeted, biased, or steered MD, importance sampling, umbrella sampling, replica exchange, local flattening of the energy surface, activation relaxation, conformational flooding, swarm methods, and others [32, 126–136]. Efficiency concerns are also addressed through coarse graining and techniques based on normal mode analysis and elastic network modeling [137, 138, 138–147]. Some methods focus on deforming a trivial conformational path (obtained, for instance, through morphing) to improve its energy profile. Examples include the nudged elastic band, morphing, zero-temperature string, and finite-temperature string methods [148–154]. While the incorporation of a suitable bias towards the goal structure forces the simulation to reduce dwell time in a given stable or meta-stable state, the bias possibly sacrifices a more expansive view of possibly different transition trajectories to the goal structure. This is typically addressed by repeating the simulation to obtain many transition trajectories, which taken together can cover the transition ensemble in the absence of correlations between trajectories.

Robotics-inspired Sampling-based Approaches

Since simulation of dynamics is the limiting factor in dynamics-based methods, efficiency concerns can be addressed by foregoing or at least delaying dynamics until credible conformational paths have been obtained. A different class of methods focus not on producing transition trajectories but rather computing a sequence of conformations (a conformational path) with a credible energy profile. The working assumption is that credible conformational paths can then be locally deformed with techniques that consider dynamics to obtain transition trajectories. Most notably, methods in this category adapt sampling-based search algorithms developed for the robot motion-planning problem which bears strong analogies to the problem of computing conformations along a structural transition.

The framework we propose in this chapter falls in the category of robotics-inspired methods and exploits analogies between the robot and molecular motion computation problems. It is worth noting that such methods predominantly model only the protein and not the other systems with which protein may interact as it switches its functional state. The foundation for this is based on the conformational selection model [155,156]. Under this model, many different functional states of a protein co-exist at equilibrium even in the absence of binding parters, albeit with different probabilities.

5.1.3 Robot Motion Planning and Molecular Motion Computation

The objective in robot motion planning is to obtain paths that take a robot from a start to a goal configuration. In robotics and molecular motion computation, a start and a goal state are specified. The goal is then to produce a feasible path that the system can follow to navigate its environment and transition from the start to the goal state. There are some unique challenges offered up by molecular systems. First, molecular systems typically have an exceptionally high number of DOFs or parameters compared to most robotic systems, hundreds or thousands of DOFs compared to at most a dozen. Second, the cost surface associated with the robot configuration space typically only has to account for the presence of obstacles and perhaps other kinodynamic constraints on the robotic system (bounds on velocities, accelerations). In a protein system, the cost surface or energy surface is typically more complex and with many local minima. While for a robotic system the question of "is a configuration feasible or not" can be typically answered deterministically, for a protein, the answer is a probabilistic estimation.

Tree-based Robot Motion Planning Methods

The methods developed in algorithmic robotics to address the robot motion planning problem fall under either the roadmap-based or tree-based category. The method we propose in this chapter falls under tree-based methods, which have proven less challenging to adapt to the molecular motion computation problem. Roadmap-based methods, which we detail in

the two following chapters in this thesis to provide a foundation for the rest of our work on motion computation, suffer from the steering problem [157, 158]; essentially, given two sampled conformations or configurations, it may not be possible to find a constraint-satisfying path steering the system from one conformation to another. Under tree-based methods, there are various ways to get around this issue (it is worth noting that the subject of the last chapter of our thesis is to equip roadmap-based methods with the ability to address the steering issue, as well). We focus here on describing the main tree-based methods in robotics and their adaptations for molecular motion computation, so we can justify the novel components in the tree-based framework we propose in this chapter.

The rapidly-exploring random tree (RRT) [159], expansive search trees [160], and path-directed subdivision tree (PDST) [161] are the three main variants of tree-based robot motion planning algorithms. They vary by how the tree is grown in the robot configuration space.

RRT RRT expands the tree by randomly sampling a new configuration Q_{rand} [159]. The closest existing configuration in the tree is located and called Q_{near}. The algorithm then expands Q_{near} in the direction of Q_{rand} using a controlled move size (potentially adding many configurations to the tree, which are at most the step-size apart in distance). In practice, it has been shown that selecting Q_{goal} as Q_{rand} with some probability p improves the performance of the algorithm. In summary, RRT expands the tree in random directions which is highly dependent on how Q_{rand} is sampled. This strategy influences RRT to explore the "frontier" regions of configuration space.

EST EST expands the tree by selecting and expanding existing configurations within the tree [160]. Briefly, a configuration c is selected using some probabilistic weighting scheme. The selection configuration is then slightly perturbed to arrive at a new configuration c'. If a collision free path can be obtained between c and c', c' is added to the tree. The performance of EST is tightly coupled to how configurations are selected for expansion. Ideally, this method would favor a uniform coverage of Q_{free}, however, this is difficult if not

intractable in high dimensional settings. Typically, a low-dimensional projection method is used to approximate the density of the samples in the tree. A new node is selected for expansion using a probability distribution over the cells that is inversely proportion to the density. This biases the growth of the tree to avoid oversampling.

PDST PDST was introduced in [161] and was designed to deal with motion problems that suffer from significant drift, under actuation, and discrete system changes. This method uses a low-dimensional projection of the sampled configurations to approximate coverage of the configuration space. This projection is decomposed into cells. All cells are stored in a priority queue based on a score (explained below). The expansion of the tree proceeds as follows. The highest scoring cell is dequeued, and a configuration from the cell is selected uniformly at random. This selected cell is then subdivided and the resulting cells are returned to the priority queue. A cell's score is primarily based on its size, with larger cells given higher priority. A perturbation function is then applied to the selected configuration resulting in a new configuration which is stored in the tree and mapped to its appropriate cell. By selecting larger cells, the search is biased towards under explored areas of configuration space.

It is worth noting that these tree-based methods are often referred to as single-query methods, as the tree grown in configuration space can only be used to find one path from a start to a goal configuration. They cannot answer multiple queries, that is the ability to find paths between different start and goal state pairs. To address multiple queries, a different tree has to be grown each time. The approach often taken, particularly in adaptations of these methods for molecular motion computation, is to rely on numerous executions in order to sample different paths even for the same start and goal query or for different queries. In contrast, roadmap-based methods can in principle answer multiple queries or be used to find multiple paths for the same query, and as such they are the subject of our investigations in Chapters 6 and 7 . We proceed now with adaptations of tree-based methods for protein motion computation and showcase their shortcomings to motivate the novel tree-based framework we propose in this chapter.

Adaptations of Tree-based Robot Motion Planning Methods for Protein Motion Computation

Tree-based methods have been used in many protein modeling problems, including loop motions [162–164], protein structure prediction [5, 46, 102], protein-ligand modeling [165], and modeling conformation changes between protein states [1, 57, 145, 166–168].

RRT-based Adaptations Transition-RRT (T-RRT) utilizes the basic RRT algorithm to explore the motions between stable states of protein systems [167]. Very small systems, such as dialanine peptide, have been modeled using both background and side chain dihedral angles as the DOFs, which results in 7 DOFs in this case. To scale the algorithm to larger systems, normal mode analysis (NMA) and elastic networks are constructed to guide the perturbation of the backbone DOFs (ϕ, ψ) [166]. As with normal RRT, a random configuration is sampled, Q_{rand}, and an expansion technique pulls the nearest node in the tree in that direction. T-RRT incorporates a reactive temperature scheme that allows it to automatically detect when the expansion of the frontier nodes are impeded by high energy barriers. The temperature within the Metropolis criterion is continually raised until the tree is able to make progress. The temperature is then lowered as successful transitions are appended to the tree. With the incorporation of NMA, this method has been successfully applied to proteins with up to 900 amino acids.

EST-based Adaptations To the best of our knowledge, EST-based adaptations have not been pursued prior to the work described in this thesis, as published under [1, 5, 45, 46, 102]. We note that the FeLTr framework we analyzed in the previous chapter for the problem of *de novo* structure prediction is an EST-based adaptation. The method we propose in this chapter, is an EST-based adaptation to compute motions between stable functional states of a protein. To summarize, EST-based methods employ a selection technique to direct the frontier of the search in under-explored areas of conformational space. Low-dimensional projections are employed to identify areas that have been over-sampled and

direct the search away from these areas. These include geometric projections and progress coordinate projections. Once a node in these tree is selected for expansion, the molecular fragment replacement technique is employed to perturb the conformation [110]. Acceptance of the expanded conformation is based on the energetic difference between the parent and perturbed conformation evaluated using the Metropolis criterion.

PDST-based Adaptations PDST has been applied to the study of the motions between protein stable states by Haspel [57]. A protein configuration is presented by its backbone atoms and the C_β atoms. The secondary structure elements of the protein's native state are identified. The primary DOFs in this setting are the adjacent residues connecting the secondary structure units. Each secondary structure element is used in the distance calculation. For each secondary structure element, we compute the angular and distance measurements to all other secondary structure elements. The same calculations are performed in the goal structure, and the differences between the two (along with a weighting term) define the distance. Sampling consists of selecting an existing conformation from the tree and applying a small random rotation to a backbone dihedral angle residing in a loop portion of the protein. This new configuration is accepted if it residues under an energetic threshold (based on the energy of the start and goal structures). The distance function defines the low dimensional projection employed by the PDST algorithm. An additional bias is used to expand the node closest to the goal structure 10% of the time. This PDST-based adaptation has been shown to produce credible information on the order of conformational changes connecting functional states of large proteins (200−500 amino acids long) [57].

Novelty and Contribution of Proposed Framework over Related Work

We propose a novel robotics-inspired tree-based method to sample conformational paths connecting two given structural states of small to medium-size proteins ranging from a few dozen to a few hundred amino acids (214 amino acids in the largest system). Instead of employing very coarse-grained representations to simplify the search space, as in some

of the related work, our proposed method models all backbone dihedral angles. The size of the search space is controlled, however, through the molecular fragment replacement technique, which allows efficiently obtaining physically-realistic protein conformations by essentially bundling together backbone dihedral angles and sampling physically-realistic moves for them. The method adapts FeLTr for motion computation, realizing in essence that the connectivity information in the conformation tree, while not important for the *de novo* structure prediction problem, is important for the motion computation problem. Adaptations here include rooting the tree at a start conformation and biasing it towards the goal conformation while enforcing coverage of the conformational space. Due to the employment of expansions and discretization layers to make decisions on how and where to grow the tree, this method can be considered an adaptation of EST and grid-based methods in robotics [158]. One objective the method seeks to meet is to reach the goal structure and so realize at least one path. Another conflicting objective is to prevent the tree from focusing only on certain regions of the conformational space and instead forcing it to maintain conformational diversity. Combined, meeting these two objectives allows balancing the exploration between progress to the goal and coverage of the conformational space so that diverse conformational paths are found and statistics can be computed over the transition ensemble.

We detail here representative results on two well-characterized systems, Calmodulin (CaM), and Adenylate Kinase (AdK). The results show that the method is effective in elucidating conformational paths on these systems. Due to the Metropolis criterion and a state-of-the-art energy function, the paths also have credible energy profiles. The employment of a tolerance region around the goal structure allows obtaining many paths from one execution of the method. In the Results section, we employ multiple executions to obtain many paths, as commonly done by path sampling methods [57, 167]. We emphasize that these paths are not transition trajectories. The conformations in them can be considered milestones during deformations of these paths into transition trajectories.

The proposed method makes an important contribution to the problem of computing

conformational paths connecting two given states of a protein. Sampling values for individual dihedral angles is not feasible on proteins more than a few amino acids to search the space connecting states sometimes more than 13Å apart. On the other hand, the work described here makes the case that one does not have to resort to very coarse-grained representations to limit the number of modeled parameters. Instead, parameters can be bundled and credible moves, extracted from known low-energy structures of proteins, can be proposed for a series of consecutive angles in order to efficiently obtain physically-realistic intermediate conformations. As we discuss in the Conclusions section, the method proposed here opens up new lines of investigation. The results in section 5.3 suggest that work in this direction is very promising for obtain credible conformational paths connecting diverse functional states of a protein.

5.2 Methods

We now proceed with details on the proposed method. We describe the local and global bias techniques that are employed, followed by investigating and controlling the impact of utilizing the molecular fragment replacement technique on path resolution. Finally, we detail our reactive temperature scheme that allows the search to cross energy barriers that may exist between the two input functional states.

5.2.1 Main Algorithmic Components of Proposed Method

The tree-based framework discussed in our work on protein structure prediction (discussed in section 4.1.2) is utilized as a starting point for devising our method. We utilize the coarse grained AMW protein representation and accompanying energy function discussed in section 4.2.2. We modify our AMW implementation in this setting to exclude the radius of gyration (Rg) term, which is utilized in most structure prediction framework to reward compact conformations. The reason for doing so is that functional structural states of dynamic proteins may not be compact. In essence, we want to allow for openings and closings of structure as needed. The molecular fragment replacement technique discussed

in section 4.1.1 is employed as the move set, as in FeLTr.

The method grows a tree in conformational space, rooted at a given start conformation. The tree grows in iterations, at each iteration expanding a selected conformation. The expansion procedure produces conformations from a selected parent conformation, and a local bias scheme is investigated to determine whether a generated conformation should be added as a child node of its parent in the tree. The selection procedure, which selects the conformation that should be extended at a given iteration, is key to balance different criteria, such as progress towards the goal and coverage of conformational space. The selection procedure employs one or more discretization layers and bias schemes over these layers to achieve one or both criteria. We refer to these as global bias schemes.

5.2.2 Node Expansion

The expansion procedure makes use of the molecular fragment replacement technique in a short MC local search that uses the Metropolis criterion. Most of our experiments detailed in the Results section employ a medium temperature, which allows the method to accept a 10 kcal/mol energy increase with probability 0.1. The Results section shows that this temperature is effective, but achieving connectivity in more complex systems can benefit from the ability to cross higher-energy barriers. Therefore, adapting the temperature as needed by certain paths in the tree to cross energy barriers of varying heights and a reactive temperature scheme is introduced and described in section 5.2.5.

Local bias in Expansion Procedure

We employ and investigate a local bias in the context of the expansion to grow the tree with conformations that improve proximity to the goal. Essentially, moves are proposed until m conformations are obtained that all meet the Metropolis criterion. The maximum number of moves attempted is l. The conformation with the lowest lRMSD to the goal is considered for addition to the tree. We analyze two different schemes, one in which the child with the lowest lRMSD to the goal structure is added to the tree (this is the no local

bias scheme), and another in which the addition is only carried out if the child's lRMSD to the goal is no higher than that of the parent (this is the local bias scheme). The local bias scheme essentially expands the tree only with a conformations that improves proximity to the goal over that of the parent. This is a greedy scheme that does not allow a path to veer away from the goal structure and possibly explore new transition routes. We compare both schemes in the Results section for how they affect the depth (progress towards the goal) and the breadth (path diversity) of the tree. While depth is measured as the lowest lRMSD to the goal structure over all paths that reach the goal region, a heuristic is introduced in the Results section to measure path diversity.

5.2.3 Selection procedure and Global bias Schemes over Discretization Layers

The selection procedure controls to a great extent where the tree grows in conformational space. Two discretization layers are considered for the selection procedure. While one is essential to the progress of the tree towards the goal, the other is considered to add conformational diversity and possibly obtain many uncorrelated paths from one execution of the method. We employ a two-layer discretization scheme. The second layer is employed as described in section 4.1.2 and promotes greater exploration of the conformational space, resulting in greater path diversity. The first layer biases the tree to grow towards the goal state and the second promote geometric diversity within the nodes of the tree. We investigate here various global biasing schemes over the first layer, which projects conformations in the tree onto a one-dimensional (1d) grid discretizing their lRMSDs to the goal. Grid boundaries are set at $[0, D]$, where D is the lRMSD between the given start and goal structures. Note that in the original FeLTr framework, the first layer projects conformations onto energetic levels. Here, one of the objectives is to reach a specific goal structure. Hence, the progress coordinate in the first layer is not energy anymore, but distance to the goal. We employ lRMSD to measure such distance and so have a meaningful progress coordinate to the goal.

Bias Schemes over lRMSD Progress Coordinate

We investigate different (global) bias schemes, as a strong bias towards selecting low-lRMSD conformations may perform well in a small system or in a particular run due to the probabilistic nature of the method and quickly drive the tree towards the goal. However, a strong bias may also lead to premature convergence to local optima and prevent the tree from approaching the goal. This is the classic depth vs. breadth issue that characterizes greedy exploration.

Different bias schemes can be naturally defined through weighting functions over levels of the 1d grid. A quadratic function, referred to as `QUAD`, can be defined to associate a weight $w(l) = 1/[1 + l^2] + \epsilon$, with level l in the grid. The function biases the selection towards levels with low lRMSD to the goal, and ϵ is set to a small value to ensure that higher-lRMSD conformations can be selected if the method is given enough time. Another weighting function, `LINEAR`, defined as $w(l) = 1/[1+l]+\epsilon$, reduces the bias. `UNIFORM` removes bias entirely, as in $w(l) = 1/[\#\text{levels}]$. A probability distribution function then associates probability of selection $p(l) = w(l)/[\sum_{levels l'} w(l^{'})]$ with a level l. Once a level l is selected with probability $p(l)$, any conformation that maps to it is selected with equal probability for expansion. We also provide the first steps towards a probabilistic combination of different bias schemes. We compare the three basic bias schemes above to $\texttt{COMBINE}_{90-10}$, which $p = 90\%$ of the time grows the tree with no selection bias (effectively employing `UNIFORM`), and $1 - p = 10\%$ of the time employs `QUAD`. The value of p can be adaptively set in a reactive scheme to balance between tree depth and breadth, and this is a direction we will investigate in future work.

Selection and Expansion

The pseudo-code for the interplay between selection and expansion in the proposed method is shown in Algorithm 5.2. First we select a level over the 1d grid over the lRSMD progress coordinate with the probability of selection dependent on the particular weighting function

used. Many conformations in the tree would correspond to the selected lRMSD level. Rather than selecting any conformation in that level uniformly at random, an additional discretization layer is incorporated that projects the conformations into a lower dimensional 3d grid based on their geometries. A weighting function over the 3d grid allows associating probability of selection to these cells. After a cell is selected, any conformation in it can be selected uniformly at random to be a parent for the expansion procedure.

Algorithm 5.2 Pseudo-code for the node selection and expansion procedure.

```
1: function SelectNode
2:     RmsdCell = SelectGridCell()          ▷ Select from 1d grid over lRMSD to Goal
3:     USRCell = SelectUSRCell(RmsdCell)    ▷ Select from 3d USR projection
4:     C = SelectFromCell(USRCell)          ▷ Uniform random from cell
5:     return(c,USRCell)
6: end function
7: procedure SelectAndExpand(T)             ▷ T is the current temperature
8:     C = SelectNode()
9:     [C',USRCell] = ExpandNode();
10:    if Valid(C') then
11:        AddTree(C,C');
12:        AddProjection(USRCell,C')
13:    end if
14: end procedure
```

5.2.4 Controlling Magnitude of Jumps in Conformational Space for Sufficient Path Resolution

The purpose of the discretization layers and the bias schemes detailed above is to promote obtaining diverse conformational paths that reach the defined goal region. There are no additional constraints requiring these path have sufficient resolution in them. There is nothing to prevent a path reaching the goal region with one or just a few conformations. This a consequence of the granularity of the moves employed to generate conformations. The fragment replacement technique can make larger jumps in conformational space compared

to using single dihedrals. However, the bundling of dihedral angles together is necessary to be able to traverse the space in a reasonable amount of time. Providing some path resolution, where possible, is appealing. Greater conformational detail along a possible transition trajectory alleviates the task for techniques that will spend their time on pursuing deformations of these paths into actual transition trajectories.

We pursue the following simple scheme to control the magnitude of the jump in conformational space in the expansion procedure. A `step_size` is then sampled from a normal distribution centered around `target_step_size`$\mathring{A}$ with standard deviation of `std_dev`$\mathring{A}$. The expansion procedure functions as before, however all candidates whose step size (lRMSD from parent) exceeds the sampled `step_size` value are discarded. Of the remaining samples, the one closest to the goal is added to the tree. This strategy provides a local bias as opposed to the global bias over the lRMSD progress coordinate.

5.2.5 Reactive temperature scheme

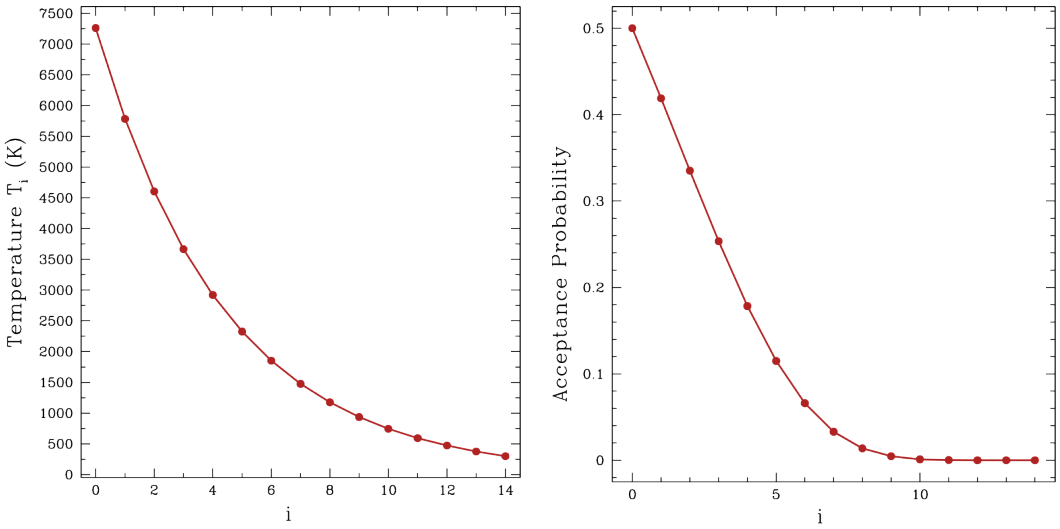
Reactive schemes that change the temperature as needed to make progress, introduced in [167] for the dialanine peptide system, present an interesting direction to further enhance the exploration of the method we propose. Building on this body of work, we investigate here a simple reactive scheme that responds to global measurements made on the conformational tree at regular intervals during the execution of the method. The progress towards the goal structure is monitored over every w iterations with no overlap (the tree grows by one conformation in each iteration), effectively sliding a window of length w over the fixed number of iterations for which the method is run. If the lowest lRMSD to the native structure by any of the conformations added to the tree during those w iterations in window i is not less than some value d_1 than the lowest lRMSD over window $i-1$, then the temperature is increased. If the lowest lRMSD achieved over window i is at least d_2 lower than the lowest lRMSD achieved over window $i-1$, the temperature is decreased.

The motivation for monitoring the tree over every w iterations is that a global decision

can be made based on the progress (or lack thereof) of all paths and their respective proximity to the goal structure. When improvements are not made, this is indicative that many paths are not able to add conformations that meet both the Metropolis criterion and extend the tree towards the goal. This means that there are energetic barriers that the current temperature does not allow crossing, and this necessitates a temperature increase. While temperature increases enhance the exploration capability, they also do not allow sufficient sampling of a local minimum by effectively increasing the magnitude of jumps that the tree makes in conformational space with every added branch. Therefore, the balance between exploration and exploitation is restored by lowering the temperature when improvements in lRMSD exceed a threshold. While large improvements may seem desirable, it is worth noting that the purpose for the method is not to quickly reach the goal structure with possibly few very long branches. Instead, the goal is to produce a series of conformations that capture the transition in some detail. Therefore, lowering the temperature effectively limits the magnitude of the jumps that the tree can make in conformational space with each branch and so provides some level of resolution in the transition from the start to the goal structure.

The temperatures considered are obtained from a proportional cooling scheme often used in the context of simulated annealing. They go from a high temperature $T_0 \approx 7261$K down to room temperature T_{14}=300K over 15 cooling steps. The fixed medium temperature used for a part of our experiments that do not employ the reactive temperature scheme corresponds to T_9. These temperatures define acceptance probabilities, under the Metropolis criterion. T_0 is defined so that the acceptance probability under it is 0.5 for an energetic increase of 10kcal/mol. T_0 is lowered 15 times according to a proportional cooling schedule that updates the temperature as in $T_{i+1} = T_i \cdot (T_{14}/T_0)^{k+1})$ until $k = 14$. The temperatures and their corresponding acceptance probabilities for an energetic increase of 10 kcal/mol are shown in Figure 5.1. This proportional cooling scheme has been employed before for simulated annealing in [48]. The reactive temperature scheme employed in this paper starts with T_9. The scheme then iterates over the temperatures. If the current temperature employed by

the method is T_i, where $0 \leq i \leq 14$, and the reactive scheme demands lowering it, then the temperature is set to T_{i+1}. If the scheme demands increasing it, then the temperature is set to T_{i-1}. The lowest temperature allowed is T_{14}, and the highest allowed is T_0.



(a) (b)

Figure 5.1: (a) Proportional cooling scheme used for the reactive temperature setting is shown. Temperatures go down from T_0 to T_{14}. (b) The corresponding acceptance probabilities, under the Metropolis criterion, are shown, using $\delta E = 10$ kcal/mol.

5.3 Results

We show the results from experiments on two systems, CaM, and AdK, of respective lengths of 144, and 214 amino acids (aa). Ten independent executions of the method are carried out on each system. The termination criterion is $10,000$ conformations. The time demands of one execution of the method is 8 hours for CaM and 24 hours of on one CPU. Multi-threaded executions of the method cut down the time requirements by a factor of 10. Energy function evaluations make up 90% of CPU time.

The tolerance around the goal structure to define the goal region is dependent on protein

size. On CaM and AdK, `tol` is set to 4 and 5Å, respectively. The maximum number of moves attempted in the selection procedure is l=100, and m=10 candidates are generated from a selected parent that all satisfy the Metropolis criterion. The target step size when controlling the magnitude of the jump in one expansion is `target_step_size` = 2.0Å. The standard deviation is `std_dev` is 0.5. The window size w used to monitor the progress of the tree in terms of lowest lRMSD in the reactive temperature scheme is set to 100 iterations. There is no window overlap. The value of the d_1 parameter defining minimum required improvement is set to 0.25Å. The value of the d_2 parameter is set to 1.5Å.

5.3.1 Experimental setup

Performance is summarized in terms of depth versus breadth. Depth is defined as the lowest lRMSD reached by the tree to the goal structure. An estimate of breadth over paths that reach the goal region is defined as $b = (\sum_{i=0}^{h} (i+1) \cdot d_i)/h$, where h is the number of nodes on the shortest path, and d_i is the maximum pairwise lRMSD among conformations at level i across all paths (i grows from goal to root). This measure downweights differences in lower levels (closer to the goal).

A total of five settings are considered: (i) only one discretization layer is used in the selection procedure, and four different bias schemes are considered over the progress coordinate. No local bias is employed in the expansion procedure; (ii) local bias is added in the expansion procedures; (iii) the magnitude of the jump in conformational space in the expansion procedure is restricted through the step size mechanism described in Methods; (iv) A second discretization layer is added over a geometric projection of the conformational space; (v) A reactive temperature scheme is considered as opposed to a fixed-temperature exploration.

On CaM, we analyze the ability to connect all 6 directed pairs that can be defined over its three functional states. These states are documented under PDB ids 1cfd (apo), 1cll (holo), and 2f3y (collapsed). CaM is an ideal system to study, as it is a key signaling protein in many cellular processes exhibiting a particularly large conformational rearrangement. On

AdK, a variety of states have been reported, but we focus here on the most studied transition from the apo/open (PDB id 4ake) to the closed state (PDB id 1ake).

5.3.2 Comparison of global bias schemes over progress coordinate

Table 5.1 summarizes performance in terms of depth, or the lowest lRMSD obtained to the goal structure from a tree rooted at a given start structure. We focus here on the first setting, where no local bias is implemented in the expansion procedure. All four bias schemes on the progress coordinate are tested in the selection procedure. Results are averaged over 10 independent executions, and Table 5.1 shows averages (μ) and standard deviations (σ) in depth across the various global bias schemes. The results obtained under `QUAD`, `LINEAR`, `UNIFORM`, and $\texttt{COMBINE}_{90-10}$ are reported in columns 3−6, respectively.

Table 5.1: Average (μ) and standard deviations (σ) are reported for the lowest tree lRMSD over 10 executions of the method. Weighting schemes for global bias over node selection are compared here. No local bias is used in the expansion procedure.

System	Start → Goal	$\mu \pm \sigma$ over lowest lRMSDs (Å) wo/Local Bias			
		`QUAD`	`LINEAR`	`UNIFORM`	$\texttt{COMB}_{90-10}$
CaM	1cfd → 1cll (10.7 Å)	3.22 ± 0.13	3.49 ± 0.42	3.69 ± 0.26	3.36 ± 0.13
	1cll → 1cfd (10.7 Å)	3.42 ± 0.24	3.66 ± 0.33	3.97 ± 0.17	3.49 ± 0.24
	1cfd → 2f3y (9.9 Å)	3.83 ± 0.43	3.76 ± 0.52	4.23 ± 0.31	4.01 ± 0.34
	2f3y → 1cfd (9.9 Å)	3.50 ± 0.26	3.54 ± 0.37	3.80 ± 0.17	3.57 ± 0.28
	1cll → 2f3y (13.44 Å)	1.76 ± 0.53	1.52 ± 0.31	1.44 ± 0.25	1.50 ± 0.20
	2f3y → 1cll (13.44 Å)	0.86 ± 0.25	0.80 ± 0.20	1.06 ± 0.31	0.94 ± 0.14
AdK	1ake → 4ake (6.95 Å)	4.20 ± 0.51	4.39 ± 0.47	5.47 ± 0.28	4.32 ± 0.41
	4ake → 1ake (6.95 Å)	4.48 ± 0.86	5.62 ± 0.80	5.94 ± 0.15	5.09 ± 0.69

Table 5.1 shows that the method effectively reaches the goal. On CaM, the average lowest lRMSDs range from sub-angstrom to slightly over 4Å, depending on the pair connected. Some pairs seem more difficult than others. On the 1cfd to 1cll paths, the average lowest lRMSDs are below 4Å, which is in general agreement with the 1.5−5Å proximity reported by MD- and MC-based biophysical studies [137, 169]. AdK represents a more challenging case

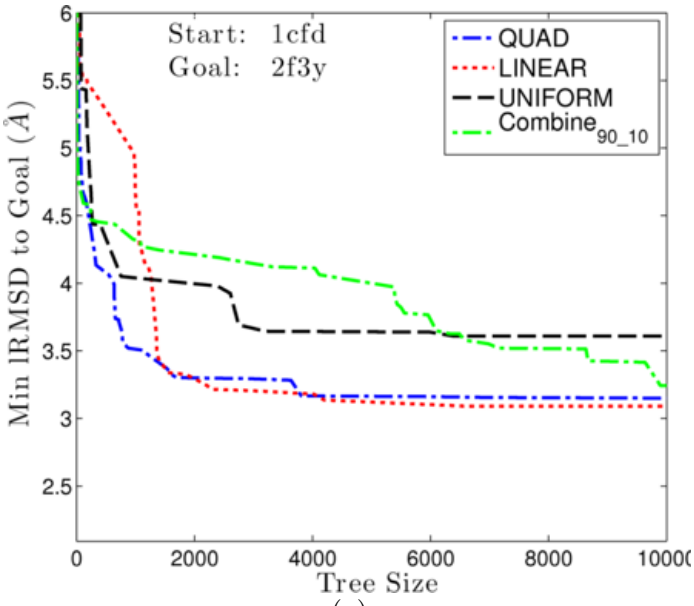
for method. Lowest lRMSDs obtained here are 4-6Å , slightly higher than the 2.5Å obtained with very coarse-grained models [57].

Results in Table 5.1 suggest that all bias schemes allow approaching the goal. Here we take a closer look at how these schemes lower lRMSD to the goal over time. We limit the analysis to one of the transitions in CaM and the "best" execution (over 10) that allows a bias scheme to achieve its lowest lRMSD to the goal structure. Figure 5.2(a) highlights the expected behavior, showing that `QUAD` can drive the exploration rapidly towards the goal but may plateau for long periods of time. `LINEAR` shows a similar rate of descent, followed by $\texttt{COMBINE}_{90-10}$. Of all bias schemes, `UNIFORM` and $\texttt{COMBINE}_{90-10}$ exhibit a more gradual decrease in lRMSD, suggesting that the exploration is more diverse under these two schemes. We recall that, while the tree is not globally biased towards the goal under `UNIFORM`, a conformation added to the tree in the expansion procedure is chosen to be the one closest to the goal among energetically-credible conformations generated from a selected conformation (this is the 'no local bias' setting). In Figure 5.2(b) we analyze path diversity or breadth on the same 1cfd to 2f3y transition. Figure 5.2(b) shows the breadth estimate across all bias schemes and confirms that diversity is higher in `UNIFORM` and $\texttt{COMBINE}_{90-10}$. Taken together, these results suggest that the $\texttt{COMBINE}_{90-10}$ global bias provides the right compromise between depth and breadth.

Comparison of schemes in expansion procedure

The experiments detailed above are repeated to measure the effect of adding a local bias in the expansion procedure (which only adds the candidate with lowest-lRMSD to the goal structure as the child of a parent node if its lRMSD to the goal is also less than that of the parent conformation to the goal) and controlling the magnitude of the conformational jumps from parent to child (described in Methods as limiting step size). In order not to add too many constraints for the expansion procedure, the step size is not controlled when incorporating local bias in the expansion procedure.

Detailed results obtained when incorporating local bias in the expansion procedure are



(a)

Bias Scheme	Breadth (Å)
QUAD	32.72
LINEAR	31.03
UNIFORM	34.44
COMBINE_{90_10}	43.67

(b)

Figure 5.2: (a) Minimum lRMSDs to the goal structure are plotted as a function of tree size and compared among global bias schemes. No local bias is employed in the expansion procedure. (b) Global bias schemes are additionally compared in terms of path diversity.

reported in columns 3−6 in Table 5.2. Overall, introduction of the local bias does not significantly improve the ability of the method to get closer to the native structure, but lower lRMSDs to the goal are obtained over the baseline setting when no local bias is implemented in the expansion procedure. On the 1cfd to 2f3 transition in CaM and vice versa, the average lowest lRMSDs are now consistently under 4Å. Slight improvements are also obtained on the AdK closed-to-open transition and vice versa.

An additional analysis shown in Figure 5.3(a) tracks minimum lRMSD to the goal over the tree during the execution of the method and compares breadth among the different global bias schemes. Similar to the results shown above for the baseline setting of no local bias, `QUAD` plateaus early. The decrease in lowest lRMSD to the goal is more gradual under `UNIFORM` and `LINEAR`. The best improvement is obtained by $\texttt{COMBINE}_{90-10}$. The comparison of breadth values in Figure 5.3(b) shows that `LINEAR` and `UNIFORM` have the highest breadth, followed by $\texttt{COMBINE}_{90-10}$. Taken together, these results suggest that adding local bias in

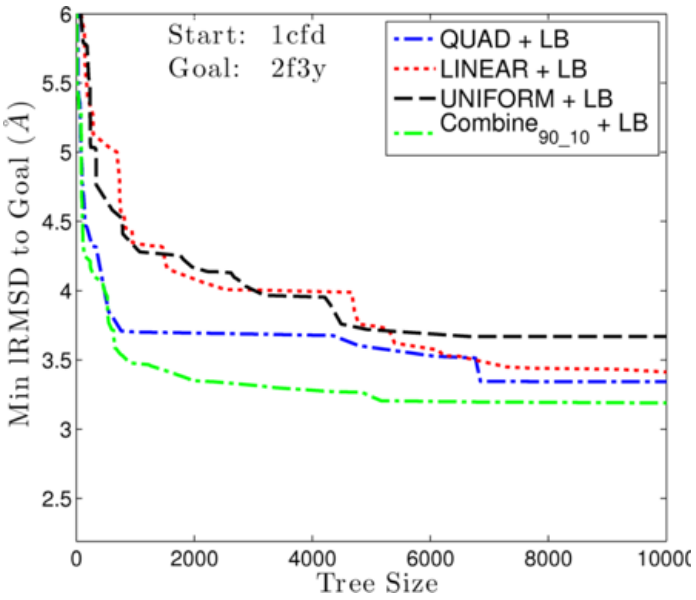
Table 5.2: Average (μ) and standard deviations (σ) are reported for the lowest tree lRMSD over 10 executions of the method. Weighting schemes for global bias over node selection are compared here. Local bias is incorporated in the expansion procedure.

System	Start → Goal	$\mu \pm \sigma$ over lowest lRMSDs (Å) w/Local Bias			
		QUAD	LINEAR	UNIFORM	COMB_{90-10}
CaM	1cfd → 1cll (10.7 Å)	3.17 ± 0.25	3.27 ± 0.10	3.49 ± 0.26	3.32 ± 0.12
	1cll → 1cfd (10.7 Å)	3.35 ± 0.51	3.56 ± 0.29	3.70 ± 0.23	3.50 ± 0.21
	1cfd → 2f3y (9.9 Å)	3.93 ± 0.37	3.93 ± 0.42	3.99 ± 0.24	3.76 ± 0.41
	2f3y → 1cfd (9.9 Å)	3.43 ± 0.39	3.65 ± 0.45	3.62 ± 0.13	3.34 ± 0.13
	1cll → 2f3y (13.44 Å)	1.91 ± 0.58	1.67 ± 0.49	2.01 ± 0.86	1.68 ± 0.37
	2f3y → 1cll (13.44 Å)	0.82 ± 0.30	0.72 ± 0.08	1.02 ± 0.43	0.73 ± 0.10
AdK	1ake → 4ake (6.95 Å)	3.91 ± 0.34	4.28 ± 0.36	5.15 ± 0.30	4.19 ± 0.22
	4ake → 1ake (6.95 Å)	4.65 ± 0.71	5.32 ± 0.79	5.62 ± 0.37	5.21 ± 0.41

the expansion procedure does not significantly improve proximity to the goal structure, but it may limit diversity. In both settings, COMBINE_{90-10} provides a compromise between depth and breadth.

Rather than adding local bias in the expansion procedure, one can try to limit the magnitude of a move from parent to child in the tree in order to provide some minimal path resolution. We now do so by limiting the size (lRMSD) of a branch from parent to child (step) as described in the Methods section. Figure 5.4 compares the distribution of step sizes in the exploration tree as a result of limiting them with the procedure described in the Methods section to the underlying distribution in the baseline setting where step sizes are not controlled. The comparison focuses on the 1cfd to 2f3y transition in CaM, employing COMBINE_{90-10} for the global bias over the progress coordinate).

Figure 5.4 allows drawing two conclusions. First, the Metropolis criterion in the expansion procedure implicitly biases step sizes even when no additional control is applied over them. Most step sizes are no more than 2Å. Second, explicitly controlling the step size as described in the Methods procedure is effective and does not significantly change the underlying distribution significantly. The procedure described to control step sizes aims to center them around 2Å, which is not very hard to do, as seen in the underlying distribution.



(a)

Bias Scheme	Breadth (Å)
QUAD	29.62
LINEAR	45.73
UNIFORM	53.28
$COMBINE_{90\_10}$	33.17

(b)

Figure 5.3: (a) Minimum lRMSDs to goal are plotted as a function of tree size and compared among bias schemes. Local bias is employed in the expansion procedure.(b) Global bias schemes are additionally compared in terms of path diversity.

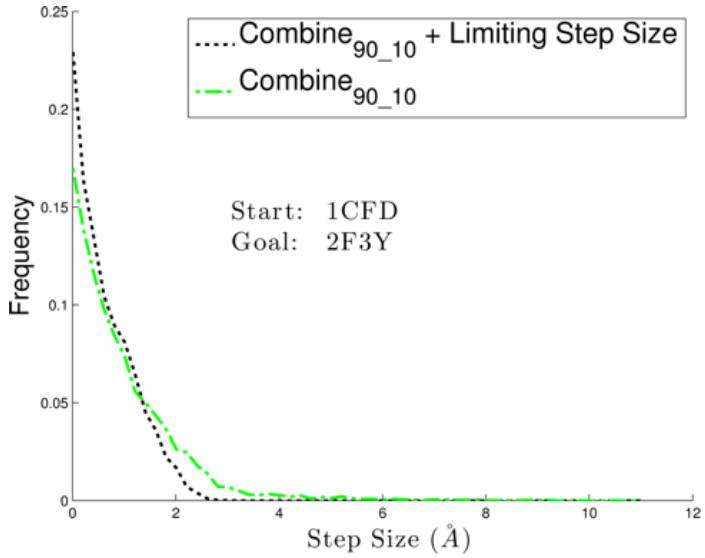


Figure 5.4: Step size is measured as the lRMSD between a parent and child in the tree structure. The distributions of step sizes in the exploration is highlighted on one selection transition for CaM, over all global bias schemes when using no local bias in the expansion procedure.

We now analyze the effect that explicit control over step sizes has on the ability to reach the goal. Figure 5.5 shows the depth reached when controlling the step size on three selected transitions of medium- to high-difficulty for the method (as determined on the baseline setting of no local bias in the expansion procedure). The best run over 10 is shown. The depths reached on each of the three selected transitions are visually compared to those obtained when not controlling the step size, whether incorporating local bias or not in the expansion procedure. Again, the best run is shown for these other settings in terms of depth. The global bias schemes considered here are either `QUAD` or $\texttt{COMBINE}_{90-10}$. Figure 5.5 shows that, when limiting the step size, it is harder for the method to achieve similar proximity to the goal structure. Most decreases in proximity to the goal are less than 1Å. A higher decrease of about 2Åis observed for the 2f3y to 1cll transition in CaM.

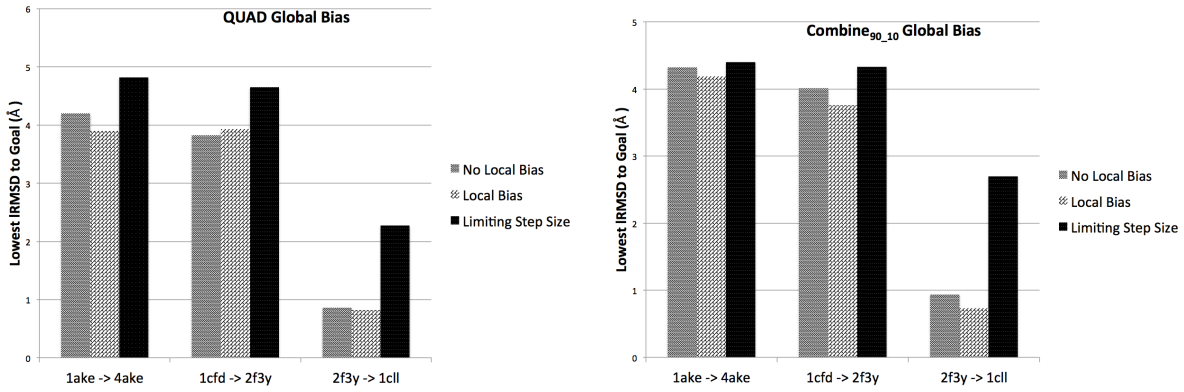


Figure 5.5: Depth is compared across the three different local schemes considered in the expansion procedure. The global bias schemes considered are (a) `QUAD` and (b) $\texttt{COMBINE}_{90-10}$. The comparison is highlighted on three selected transitions.

The adverse effect on proximity when limiting the step size is expected. Demanding more resolution along conformational paths in the tree distributes more conformations and computational resources to obtaining more intermediate points along a path rather than extending paths toward the goal structure. Increasing the size of the conformational tree

allows obtaining similar depth when controlling the step size to the other two settings. This is observed when running the method to sample $25,000$ rather than $10,000$ conformations (data not shown).

5.3.3 Analysis over incorporating geometric discretization layers

In this setting we add the second USR-based discretization layer, thus discouraging the tree from visiting the same regions in conformational space too often. As discussed in the Methods section, this is achieved by projecting conformations in the tree onto a 3d grid. We limit the analysis here to the setting of using the $\texttt{COMBINE}_{90-10}$ global bias scheme over the progress coordinate for the first discretization layer and employing no local bias in the expansion procedure. Figure 5.6 compares the depth (top row) and breadth (bottom row) of the tree obtained when incorporating the geometric projection layer as opposed to not incorporating it. The shown values for depth and breadth correspond to the run that achieves the best depth (lowest lRMSD to the goal) over 10 runs.

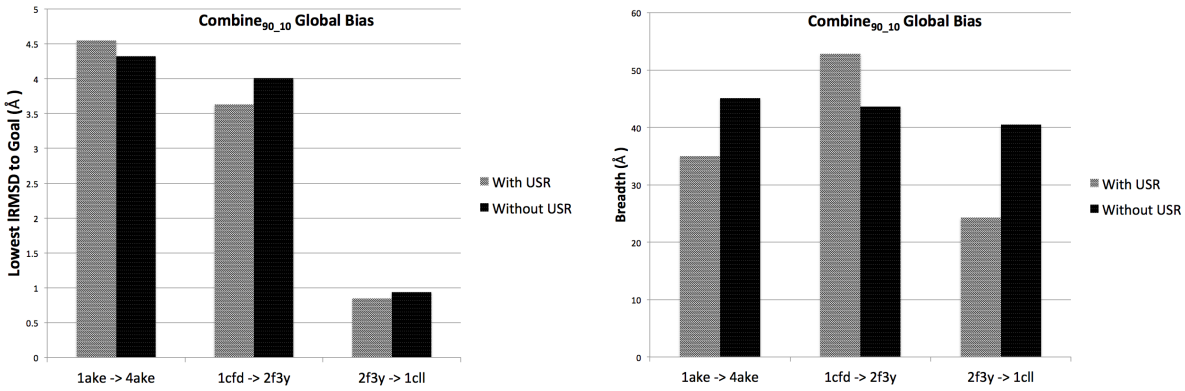


Figure 5.6: Depth (a) and breadth (b) are compared when using the second discretization layer ('with USR' in legend) over not using it ('without USR'). The 'without USR' setting is the baseline setting where no local bias is employed in the expansion procedure. The global bias scheme considered here is $\texttt{COMBINE}_{90-10}$. The comparison is highlighted on the same three selected transitions.

The comparison shows that insisting on diversity does not significantly hamper the tree from reaching the goal structure with similar lowest lRMSDs. Differences in depth are within 0.5Å. In fact, on two transitions, slight improvements are obtained. It is important to note that the extent of improvements of depth depends both on the extent of sampling and on whether paths need to be fine tuned or altogether alternative routes have to be found. When fine tuning is needed, a finer granularity in the 3d grid for the geometric projection of the conformational space may help further improve proximity to the goal. Comparison of breadth values shows that the improvements in breadth and depth are correlated. This is a consequence of the fact that, when the projection layer increases lRMSD to the goal, fewer paths are considered successful and counted in the breadth analysis.

5.3.4 Analysis over incorporating reactive temperature scheme

All of the above experiments employ a fixed temperature corresponding to T_9 in the temperature schedule shown in the Methods section. Here we consider a reactive temperature scheme, as described in the Methods section, to enhance sampling and allow paths to jump over energy barriers as needed. In this setting, we set the maximum number of moves attempted in the expansion procedure to l=250, and m=25 candidates are generated from a selected parent that all satisfy the Metropolis criterion. When increasing the temperature, the exploration is more likely to yield conformations farther in conformational space, and so it is harder to obtain children that approach the goal. The higher number of moves and children allow us to address this.

Figure 5.7 shows the depth reached when incorporating the reactive temperature scheme on the same three selected transitions of medium- to high-difficulty for the method. The best run over 10 is shown. The global bias scheme employed over the progress coordinate is $\texttt{COMBINE}_{90-10}$ (only one discretization layer is used), and no local bias is used in the expansion procedure. The depths reached on each of the three transitions are visually compared to those obtained when employing a fixed temperature (T_9) instead of the reactive scheme. Figure 5.7 shows that the reactive temperature improves depth for all three transitions.

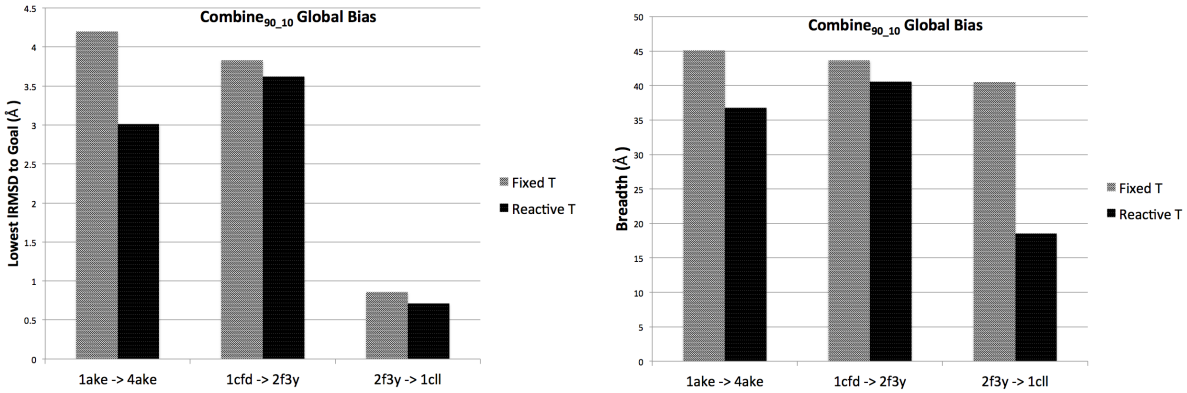


Figure 5.7: These graphs illustrate the effects of our reactive temperature scheme. This illustrates that while we sacrifice some of the breadth of our search tree, the reactive scheme is able to locate conformations closer to the goal state. This is more pronounced for the larger system (AdK).

Further analysis of depth shows that the reactive temperature scheme provides the best improvement, by more than 1Å in the case of AdK. This transition is difficult, and the improvement in depth by allowing paths to cross energy barriers suggests that the transition goes over high-energy regions. In the other two transitions, where the baseline setting of the method achieves good proximities to the goal structure, the reactive temperature scheme offers slight improvements in proximity to the goal. Breadth is also higher, which suggests that more paths are sampled by the method when allowed to jump energy barriers.

5.3.5 Detailed analysis on CaM transition ensemble

On CaM, the method is able to surpass initial lRMSDs >13.44Å. Sub-angstrom lRMSDs are obtained when the method is setup to approach 1cll from 2f3y; $1-2$Å are obtained in the other direction. Connecting the other 4 directed pairs is more difficult; lowest lRMSDs across all bias schemes are $3-4$Å. The employment of USR-based discretization seems to slightly improve the lowest lRMSDs in these difficult cases.

Results on CaM are in qualitative agreement with those observed in experiment and simulation [137, 170, 171]. The transition between 1cll and 2f3y is easier than between the

other pairs. Though the other pairs have initial lRMSDs that are lower than that between 1cll and 2f3y, the true distance that has to be surpassed is in angular space, which partially explains why the method performs well. Due to its use of molecular fragment replacement, the method is particularly suitable to obtain paths of "angular" rearrangements. Some paths highlighting these rearrangements are shown in Figure 5.8.

We note that the use of fragment configurations is justified when functional transitions do not involve unfolding. This is true of many proteins, including CaM and AdK. In particular, wet-lab studies on CaM wild types and mutants exclude the possibility that the transition involves a significant population of unfolded or disordered states [171]. These studies also suggest that the transition between 1cfd and 1cll is a complex process with energy barriers rather than a single global transition between two substates. A pseudo-free energy landscape produced by our method is shown in Figure 5.9. All paths from 10 runs obtained with $\texttt{COMBINE}_{90-10}$ and local bias in the expansion procedure on connecting 1cfd to 1cll and vice-versa are combined. Pseudo-free energies are calculated along the ΔR coordinate (defined as lRMSD(C, C_{1cfd}) - lRMSD(C, C_{1cll})) through the weighted histogram analysis method [172]. The pseudo-free energy landscape in Figure 5.9 shows that paths have to cross regions of high free energy, which qualitatively agrees with wet-lab findings in [171]. The shown pseudo-free energies need to be taken with caution. Pseudo-free energy values are affected by potential lack of sampling density and path diversity.

5.3.6 Detailed analysis on AdK transition ensemble

The transition from the closed (corresponding to PDB id 1ake) to the open state (PDB id 4ake) in AdK has been the subject of many recent studies. We show in Figure 5.10 a sample path capturing the conformational change from the closed to the open structure. This path, which reaches the goal structure with an lRMSD of less than 3Å, is the best one in terms of depth obtained with the reactive temperature scheme, using $\texttt{COMBINE}_{90-10}$ as the global bias scheme over the progress coordinate, and using no local bias in the expansion procedure. This path shows the opening of the two domains in the structural transition.

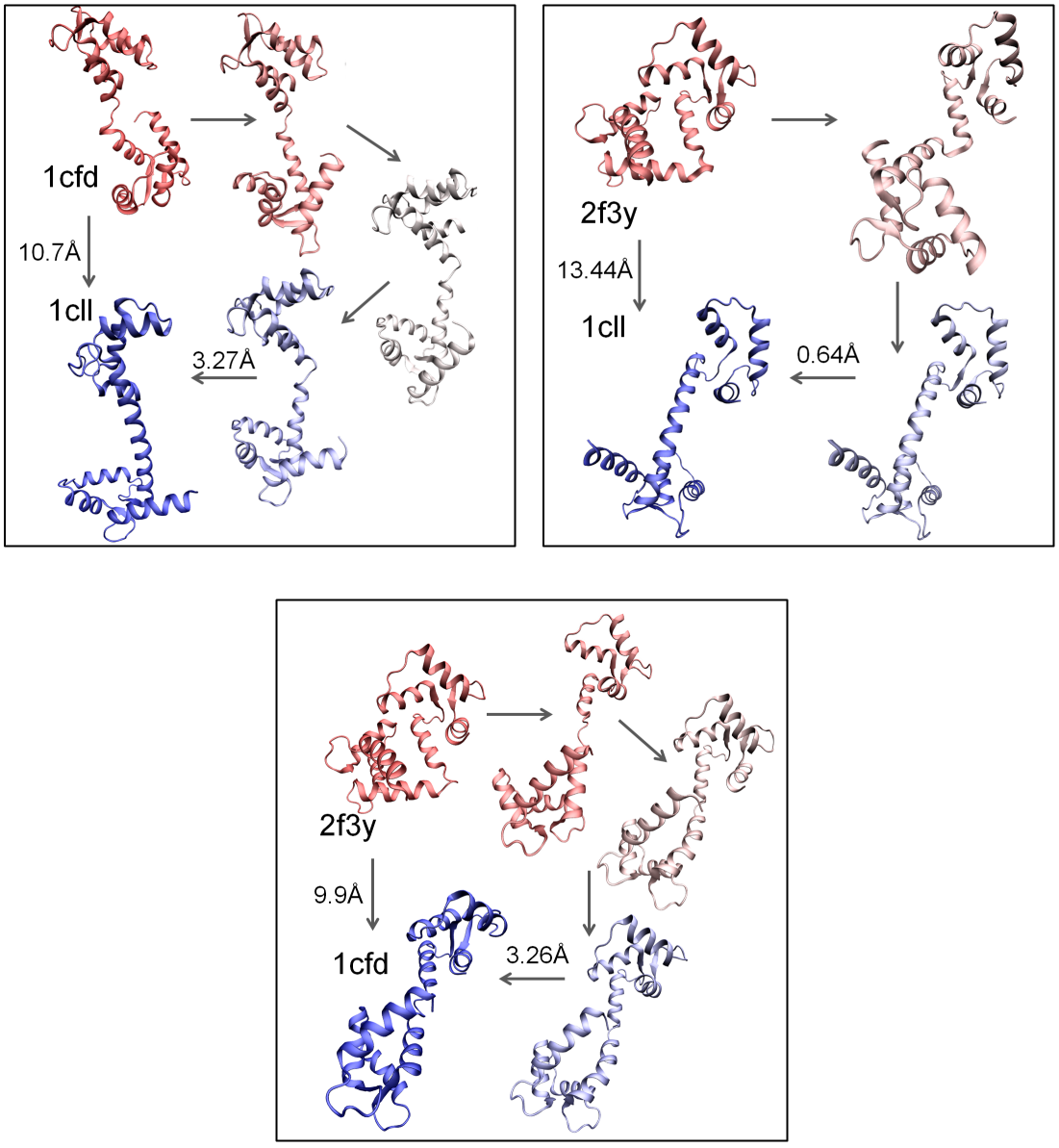


Figure 5.8: Three paths for CaM are highlighted. Start and goal structures are in red and blue, respectively. Selected conformations in the path are drawn in a red-to-blue interpolated scheme.

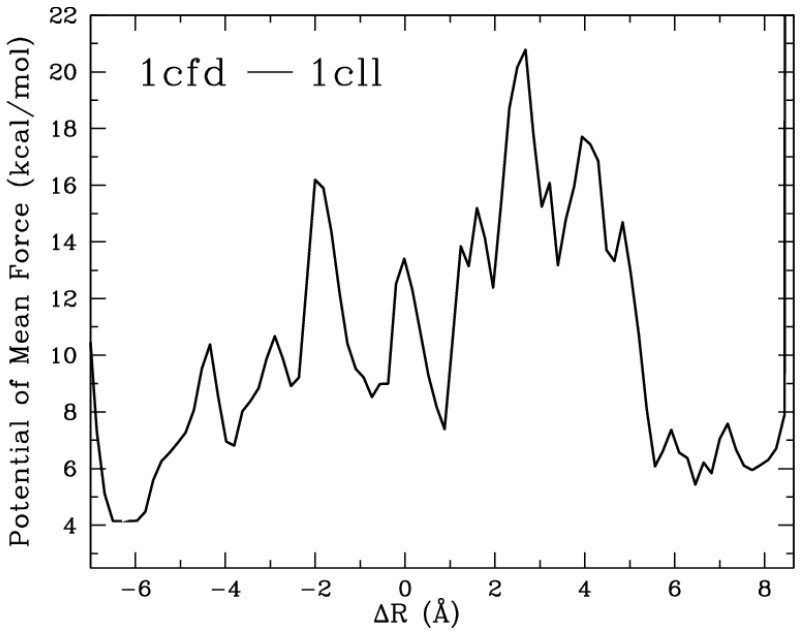


Figure 5.9: Pseudo-free energies along ΔR are shown for sampled paths connecting 1cfd to 1cll and vice versa.

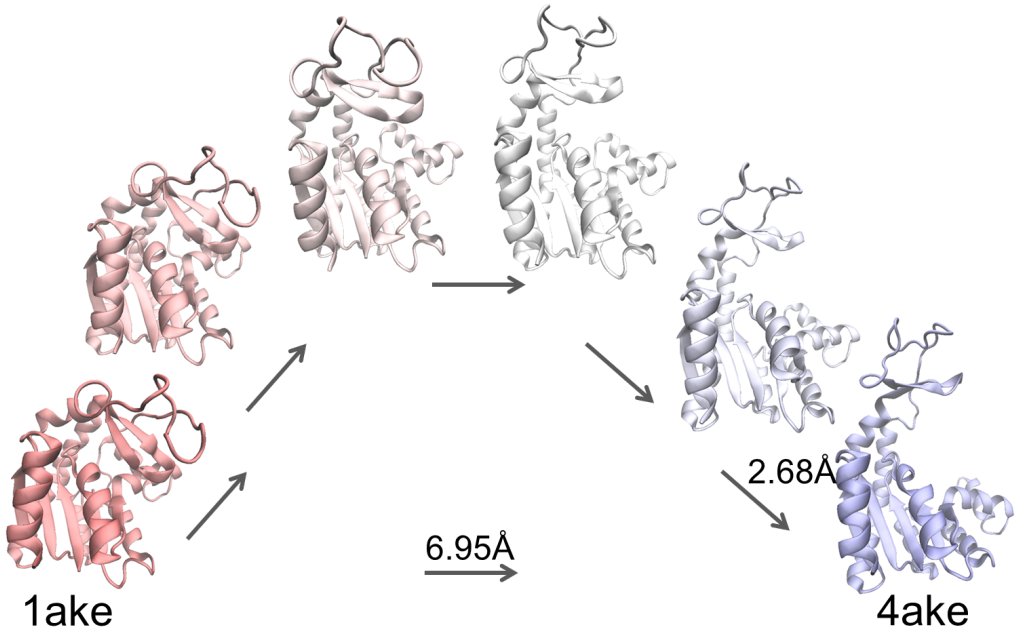


Figure 5.10: A path capturing the transition from 1ake to 4ake is shown here. Start and goal structures are in red and blue, respectively. Selected conformations in the path are drawn in a red-to-blue interpolated scheme.

Studies on modeling the closed to open transition reproduce the presence of many known intermediate structures [32, 57, 173]. In particular, in [32], all known crystal structures are analyzed for their presence in the closed to open transition. Here we conduct a similar analysis after collecting all intermediate structures deposited for AdK in the PDB. Some of these structures have been captured on systems with slight sequence variations (due to the experimental technique extracting them from different species). As in [32], we employ the SwissView homology-modeling server [174] to thread these structures onto the amino-acid sequence of 1ake so that a direct analysis can be performed in terms of lRMSD.

We measure the extent to which we find each of the 27 crystal structures as intermediate conformations (in terms of lowest lRMSD) over all paths that reach the goal within 3.5Å. We limit the analysis to the above setting of employing the $\texttt{COMBINE}90-10$ global bias scheme, using no local bias for the expansion procedure, and incorporating the reactive temperature scheme. We report the minimum lowest lRMSD over the best path over all runs (best in terms of depth). Table 5.3 reports two minimum lowest lRMSDs per structure, one for the 1ake to 4ake transition and the other for the 4ake to 1ake transition. The PDB ids of the crystal structures are shown in column 1. The ordering is indicative of the location of these structures in the 1ake to 4ake transition (structures listed at the top are closer to 1ake, and those at the bottom are closer to 4ake). Some of the known intermediate structures are in dimeric configurations in the crystal (chains A and B are available in the PDB), but we employ here only chain A for analysis, since the chains are structurally identical. Table 5.3 shows that the paths in the 1ake to 4ake transition capture most of the known intermediate structures with lowest lRMSDs below 3Å, which suggests that the method captures well the presence of known intermediates and is able to model the 1ake to 4ake transitions in AdK. On the reverse transition, the higher lRMSDs indicate that there are possibly high local maxima that limit the exploration capability and the quality of paths.

This preliminary study on AdK is promising However, AdK presents an extremely challenging case for our method, not only due to its size but also due to the presence of a significant energy barrier in the transition [129]. Tables 5.1 and 5.2 show that the lowest

Table 5.3: The lowest lRMSD to each of the known crystal structures for AdK is calculated over all paths that reach the goal within 3.5Å. The value shown in column 2 is the minimum lowest lRMSD obtained over the best run (in terms of depth) of the method using the $\texttt{COMBINE}_{90-10}$ global bias scheme over the progress coordinate, no local bias for the expansion procedure, and the reactive temperature scheme for the 1ake to 4ake transition. Column 3 shows the minimum lowest lRMSD for the 4ake to 1ake transition. Column 1 shows the PDB id of each of the crystal structures considered. The structures are ordered according to their locations along the 1ake to 4ake transition.

PDB id	lowest lRMSD (Å)	
	1ake→4ake	4ake→1ake
1e4v	0.32	3.17
1e4y	0.93	3.02
2eck	0.35	3.17
1ank	0.48	3.13
1zin	2.24	2.76
1zio	3.75	3.19
1zip	2.09	2.79
1s3g	1.71	3.31
1p3j	1.48	3.37
2eu8	1.31	3.04
2p3s	1.47	3.30
2oo7	1.26	3.06
2ori	1.30	3.04
2osb	1.31	3.07
2rh5	3.82	2.82
2rgx	2.93	3.34
1aky	1.44	3.12
2aky	1.30	3.30
3aky	1.41	3.14
1dvr	2.91	3.02
1zak	2.81	3.99
2ar7	3.65	3.10
2bbw	3.79	3.38
2c9y	3.17	3.73
1ak2	2.94	4.04
2ak2	2.89	3.97
2ak3	3.78	3.73

lRMSDs can be above 4Å to the goal structure. Lack of density in sampling makes a pseudo-free energy analysis premature for AdK. In addition to more sampling, complex proteins, such as AdK, may present additional challenges in silico possibly due to a more complex energy surface. The above analysis of the effect of the reactive temperature scheme shows that proximity to the goal structure can be improved when the temperature is changed by the method as needed for paths to cross energy barriers. This suggests that the energy landscape of AdK is complex, with transition states of potentially high energies.

5.4 Conclusions

This chapter has described a novel method to compute the conformational paths that connect pairs of known functional states of a protein system. This method combines an EST-based approach coupled with molecular fragment replacement. For the protein systems showcased here, the analysis shows that the method is capable of producing energetically-credible conformational paths connecting the known states.

Chapter 6: Modeling Protein Structural Transitions with a Roadmap-based Robotics-inspired Method: Of Stochastic Roadmaps and Markov State Models

This work described in this chapter is based on work published in a workshop [175]. Here we advance our treatment of protein motions by building on roadmap-based methods in robotics and drawing analogies between conformational roadmaps and markov state models (MSMs). In the previous chapter we introduced a tree-based robotics-inspired method devised to quickly determine a conformational path between two structural states. We recall that this was accomplished by strongly biasing the growth of the tree in conformational space. This feature, while beneficial in expediently providing a conformational path, makes it hard to obtain an ensemble of paths from one execution of the method. Due to the strong bias, even multiple executions of a tree-based method are expected to result in a path ensemble with high inter-path correlations. For these reasons, we decide to investigate roadmap-based methods. These methods, which we describe in detail below, essentially capture the connectivity of the conformational space through a graph or roadmap. The roadmap can then be queried for one or more paths connecting two given structural states. In this chapter, we describe a preliminary adaptation of such methods to handle various known algorithmic issues that are enhanced in severity in the protein modeling domain. What we actually build is a stochastic roadmap, with probabilistic edges. Moreover, we recognize and exploit analogies between the stochastic roadmap and MSMs to extend the analysis from path querying to obtaining interesting statistics on structural transitions. The latter allows us to compare variants of a protein molecule and, specifically, to explain the impact of mutations on known oncogenic proteins.

6.1 Background and Related Work on Roadmap-based Methods

6.1.1 Probabilistic Roadmap

The Probabilistic Roadmap (PRM) method was introduced in [176] for the robot motion planning problem. The method consists of two stages, the *learning* phase and the *query* phase. The first phase builds a representation of the obstacle-free robot configuration space in a graph/roadmap, whereas the second phase queries the roadmap for paths. The query phase typically relies on known graph search algorithms to produce a lowest-cost path. As we detail below, in a roadmap where edges are probabilistic, the query can be used to sample paths, as well, essentially replacing the need to launch MD or MC trajectories. The main challenges with PRM lie in the learning phase. In the original PRM, each vertex, V_i in the roadmap represents a free configuration of the robot. Each edge, E_i, encodes a collision-free path between two vertices. The roadmap is built in two steps, a construction step and an expansion step. The construction step starts by randomly generating configurations in the free configuration space. Each configuration/vertex is then connected to a pre-determined number of nearest neighbor configurations/vertices in the roadmap. Edges that contain collisions with obstacles are then removed. The result of this process is rarely a connected graph that can be used to answer queries. For this reason, the expansion step is pursued to further populate the roadmap with configurations in regions deemed critical to bridge connected components. We note that an effective sampling procedure is critical to the success of this method but very challenging. Biased sampling techniques have been proposed over the years, particularly to focus sampling on difficult narrow regions in the free robot configuration space. However, much work remains to be done when transferring PRM to the protein conformational space. It is highly improbably that a conformation sampled uniformly at random will be energetically feasible. Designing an effective move set is thus key. Moreover, the protein conformational space is vast. It is important to focus sampling in the vicinity of a particular start and goal structure pair while minimizing bias.

None of these issues are currently well explored, and they are the motivation for the work proposed in the remainder of this thesis.

6.1.2 PRM Application in Protein Modeling

PRM has been adapted and applied to many protein modeling problems. Empirical energy functions replace collision checking during the *learning phase*. Nearest neighbor calculations typically employ lRMSD or L1 norm over backbone dihedral angles. Early work applied PRM to modeling the docking of small, flexible ligands onto a rigid protein molecule [177]. The receptor protein was held stationary, and 6 DOFs were provided to translate and rotate the ligand relative to the protein. A few additional dihedral angles were modeled in the ligand to allow it to flex and improve its energetic interaction with the protein. The sampling of configurations was biased towards lower-energy configurations as the number of samples increased.

Work in [178,179] applied PRM to analyzing protein folding. Given the native structure of a protein, this application of PRM discovered paths from random unfolded conformations to the goal native structure of the protein. A novel technique generated configurations. A set of Gaussian distributions were employed to perturb native values for the dihedral angles, using increasing standard deviations in order to allow moving further away from the native structure towards unfolded conformations. Sampling was terminated when conformations were sampled that contained zero native contacts.

6.1.3 Stochastic Roadmap Simulation (SRS)

Work in [180] extended the treatment and proved that a carefully-constructed (stochastic) conformational roadmap converges in the limit to the same distribution as MC samples. The difference with prior adaptations of PRM is that edges in the roadmap are now probabilistic, encoding transition probabilities between two vertices in the roadmap. Transition probabilities are calculated as shown in Equation 6.1.

$$P_{ij} = \begin{cases} (1/|N_i|)\ exp(-\Delta E_{ij}/k_b T) & \text{if } \Delta E_{ij} > 0 \\ (1/|N_i|) & \text{otherwise;} \end{cases} \tag{6.1}$$

$$Pii = 1 - \sum_{i \neq j} P_{ij} \tag{6.2}$$

In this equation, $|N_i|$ measures the number of neighbors (or out-degree) of the vertex v_i, k_B represents the Boltzmann constant, and T is the temperature. The ΔE_{ij} term refers to the difference in energy between vertices v_i and v_j, ΔE_{ij} = E(V_j) - E(V_i). A self transition probability normalizes the sum of all the probabilities to one. The stochastic roadmap in [180] calculated folding rates by solving a set of linear equations derived from the transition probabilities, effectively avoiding having to collect statistics on a large numbers of random walks, as would be done if employing MD or MC to simulate folding events.

6.2 Methods

We pursue here an adaptation of the stochastic roadmap to model structural transitions in medium-size proteins. Our method consists of three stages. First, we generate samples or conformations utilizing an evolutionary algorithm developed in the Shehu Lab [181]. We then organize these conformations into structural states and a roadmap is constructed to encode the connectivity among these states utilizing a "lazy" local planner. Constructed over states, the roadmap is a Markov State Model (MSM), allowing rigorous methods to be used to extract information regarding structural transition rates in addition to answering path queries. This method balances the computational effort (employing a simple local planner) and the information gain provided by the analysis of the resulting MSM.

6.2.1 Sample Generation

As noted, a key challenge in roadmap-based methods is sampling. In this work, we utilize an evolutionary algorithm (EA) developed in the Shehu lab [181]. EAs are particularly useful for hard stochastic optimization problems, and we utilize such an EA here to sample low-energy conformations of medium-size proteins. However, our domain of applicability is rather limited in this chapter. We focus on proteins for which there are many experimentally-determined structures in the PDB, and use these structures to define the search space and the move set, thus circumventing the issues that hamper applicability of PRM for modeling structural transitions. A detailed description of the EA sampling algorithm is beyond the scope of this thesis. However, we summarize here its main ingredients to understand the advantages offered by employing an EA in the learning phase.

The EA utilizes the CA trace of each experimental structures to define a low-dimensional embedding via principal component analysis (PCA). The top m principal components (PCs) that capture no less than 90% of the total variance are then utilized to define a lower-dimensional space for exploration. The EA's initial population consists of p experimental structures. Reproductive operators are utilized to add new children to the population. These operators perturb existing samples in the space of PCs and result in a new structure consisting of an m dimensional vector. The fitness of each child is determined using a multi-scaling procedure that transfers this m dimensional vector into an all-atom conformation. For each child, the CA coordinates are recovered using the m PCs, the full backbone is then reconstructed with backbone reconstruction techniques, and finally the side-chain atoms are added and the entire all-structure is optimized via Rosetta's *relax* protocol. The fitness of each conformation is its all-atom energy (determined using Rosetta's $score12$ all-atom energy function). The ensemble Ω consists of all the structures generated during the course of the EA and are fed to the next phase of our method.

6.2.2 Structural State Identification

Each vertex in our roadmap represents a structural state rather than a single conformation. The ensemble Ω may contain geometrically-similar conformations as a byproduct of the EA. We proceed to group the ensemble Ω into a collection of states, which will allow us to treat the roadmap as an MSM. Our definition of a state is that of a cluster of geometrically-similar conformations. We employ the leader clustering algorithm [182] to compute clusters/states.

The leader clustering algorithm has the benefit of not having to specify the number of clusters/states a priori. Its results are dependent on the order in which the data is processed. Here we use a sorted order, ordering first all the conformations in the ensemble by their Rosetta score12 energies. This ordering allows the first conformation mapped to a new cluster to be the lowest-energy one over all others that will be mapped to that same cluster. The algorithm proceeds in the sorted order, mapping a conformation to one of the existing clusters if its distance to the cluster representative is below a specified cluster radius. Otherwise, a new cluster is created with the unmapped conformation as its representative. The algorithm proceeds until all conformations have been processed, resulting in a list of clusters/states. The decision on what distance function to use is important. Here we employ lRMSD over only CA atoms; that is, we use CA lRMSD. We experiment with different values of cluster radii, as presented in the Results section.

6.2.3 Roadmap Construction

Our roadmap is a directed graph $G = (V, E)$. Cluster representatives identified above are used to populate the vertex set V. Edges are added using the following process. For each $v \in V$, we identify the k nearest-neighbors (k_{nn}) that are within an lRMSD distance constraint ϵ_{nn} of v. For each identified neighbor u that passes both of these conditions, we add an edge in both directions, (u, v) and (v, u), to $E(G)$. At the completion of this process, we improve the connectivity of the graph by calculating its connected components (CCs) and add additional edges to the graph (subject to ϵ_{nn}) to merge CCs.

Edges are assigned weights following the original SRS formulas shown in Eq. 6.1. Energy

values utilized in these equations are taken from the Rosetta all-atom *score*12 function (which was calculated as part of the sampling procedure). Each cluster/state is assigned the energy value of its conformation representative. We substitute the scaling parameter α in place of the K_B term in the original equation. The reason for this is as follows. The Rosetta energy function combines both physics- and knowledge-based terms. We calibrate the value of α by utilizing Rosetta's *relax* protocol, which use a stochastic method to minimize a protein structure. We perform relaxations over the set of crystal structures provided as start and goal and calculate the variance between each minimization. In the case of the Ras protein studied here, we observe a variance of $6-7$ energy units. Utilizing a statistical mechanics treatment, structures within the same energy basin should exchange into each other with high probability. Let us refer to the latter as a target probability t_{prob}, which can be a user parameter. Solving the equation $e^{-6/\alpha} = t_{\text{prob}}$ for α gives us the value to use in lieu of K_B in equation 6.1. We note that the actual value for α is dependent on the energy function employed and requires that a target probability be specified, but the process is general.

Each edge in this constructed stochastic roadmap G now encodes a potential transition between two structural states. In this work, we employ a "lazy" strategy that avoids the computation of these transitions and the steering issue in PRM, thus focusing on the global connectivity. This has some similarities to the Lazy PRM method [183]. We note, however, that foregoing a local planner is made possible here because of the stringent criterion of structural proximity when considering connecting two vertices via an edge. This in itself exploits the dense structural sampling afforded by the EA employed in the sampling stage.

6.2.4 Roadmap and Markov Analysis

By construction, our roadmap G consists of a set of strongly connected components. As demonstrated in [180], a discretized version of a Monte Carlo (MC) trajectory can be achieved by performing a random walk in G. By performing a large number of random walks, we can derive statistics related to transitions rates between states and study the differences

in realized pathways between states. Given the high variance that would result from most encodings of G, this would require a very large number of random walks to be performed. Treating the constructed stochastic roadmap as a graph allows using path search algorithms to obtain paths connecting structural states of interest. Treating the roadmap as a Markov state model allows using transition state theory to obtain measurements approximating kinetic quantities of interest.

Querying the Roadmap

As demonstrated in the original PRM method in [157], the roadmap can be queried given two states of interest. Dijkstra's algorithm can be used to obtain a shortest path. The roadmap's edges are weighted by probabilities of transition, so we take the negative logarithms of these probabilities and use these values in calculating the lowest-cost path. In addition to such a path, more information can be obtained by analyzing not just one path but several. Yen's K-shortest paths algorithm [184] can be employed for this purpose.

Treating the Roadmap as a Markov State Model

The roadmap G can be treated as an MSM encoding the stochastic behavior of the system being studied. In this work, we use the roadmap to model the structural transitions between functionally-relevant states of a protein and understand how these transitions are affected by sequence mutations. For this purpose, the roadmap G is analyzed to determine the expected number of transitions employed by a protein system to switch from one structural state to another.

Recall that structural states are vertices in the vertex set V in our roadmap G. For each vertex $v_i \in V$, one can utilize first-step analysis theory to measure the expected number of transitions t_i from vertex v_i to some specific vertex of interest. As demonstrated in[180], random walks need not be performed to obtain such a measure, as a closed-form solution can be computed via a linear solver. The formulation of t_i is recursive. Let us generalize and state that the goal is to measure the expected number of transitions from some vertex

v_i to a set of vertices $v_j \in A$, where A is a subset of V that does not include v_i. Then, provided that v_i and A are in the same SCC:

$$t_i = 1 + \sum_{v_j \in A} P_{ij} \cdot 0 + \sum_{v_j \notin A} P_{ij} \cdot t_j \quad \forall\ v_i \notin A$$

This results in a system of equations that is the same order as the number of vertices in the SCC. Since clustering of structures into structural states reduces the number of vertices in the roadmap, an exact solver (as opposed to a slow-converging iterative solver) can be afforded, and that is what we employ in this work to solve the linear system above algebraically and obtain t_i for all the vertices simultaneously.

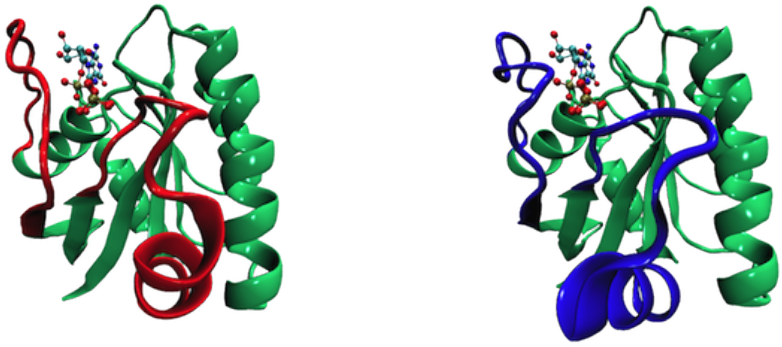
In this work, we are specifically interested in measuring the expected number of transitions from an ON to an OFF state and vice versa, with these two states denoting specific structural states critical to the ability of the Ras oncogene to function normally. By repeating the sampling, clustering, roadmap construction, and its analysis on different sequence variants of RAS, we then are able to compare the expected number of transitions between these two states of interest in the wildtype versus disease-participating variants of Ras.

6.3 Application on the RAS Oncogene

6.3.1 Experimental Setup

Here we present results on the application of the proposed method on the wildtype and Q61L variant of the Ras oncogene. Ras is a well-studied protein that regulates cell proliferation and whose variants which deregulate activity are involved in over 25% of human cancers [185]. The native activity of Ras is to switch between an ON/reactant (GTP-bound) and an OFF/product (GDP-bound) state. These two states have been characterized in the wet laboratory and can be found under structures with PDB ids 1qra and 4q21, respectively. We show these structures side by side in Figure 6.1. The CA lRMSD between these structures is 1.5Å, but changes are largely localized on two loop regions, switch I and switch II (which

our previous analysis of PCA for Ras is able to capture [186]). How variations in the Ras sequence affect its capacity for switching between states is the focus of much research and is the reason we apply our SRS-based algorithm.



(a) (b)

Figure 6.1: Left: A representative of the ON (GTP-bound) state of Ras (PDB id: 1qra). Right: A representative of the OFF (GDP-bound) state (PDB id: 4q21). The reactant (GTP) and product (GDP) are shown, as well. The two loop regions that undergo a structural change in the ON to OFF transition and (reverse) are shown color-coded in red (left) and blue (right).

The reduced space over which the sampling stage operates is obtained via PCA on 46 (wildtype and variant) structures extracted for Ras from the PDB (details on the data collection step can be found in [186]). Our method is run twice, once on the wildtype sequence and once on the disease-participating variant (Q61L). It is important to note that, while the PCs are the same in each setting, the EA algorithm obtains different structural ensembles, as the initial structures are threaded onto the sequence of study, and thus mapped by the multiscaling procedure to minima of different sequence-dependent energy surfaces. Thus, the results of the our method are dependent on the sequence used and can be used to draw comparisons between the wildtype and variants to determine how sequence mutations affect transitioning between the ON and OFF states.

The structure in the PDB entry 1qra is considered a representative of the ON state of Ras, whereas 4q21 is a representative of the OFF state. These PDB-obtained structures are each minimized 500 times with the Rosetta relax protocol (the protocol is stochastic), and the resulting structures are added to the Ω ensemble. After the clustering, the cluster containing the most minimized structures of 1qra is labeled the ON state, whereas the cluster containing the most minimized structures of 4q21 is labeled the OFF state.

6.3.2 Roadmap Analysis on Ras Wildtype and Q61L Variant

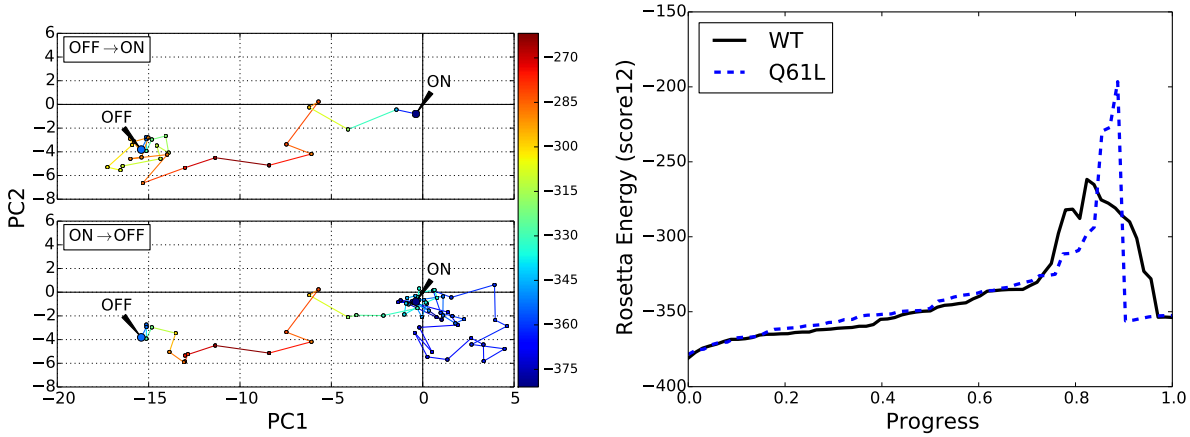
We apply the analysis techniques discussed in section 6.2.4 to the roadmaps created for the wildtype and Q61L sequences. The lowest-cost paths between the ON and OFF states for each sequence are computed and analyzed first. Column 3 in Table 6.1 shows the minimum cost of each of these paths. The cost of a path is computed as $\sum_{e=(u,v)} -log(exp(-\frac{E(v)-E(u)}{\alpha})$, where u and v are the two states connected by an edge, and we take the sum over all edges in the path, and α is the scaling term discussed in section 6.2.3.

Comparison of these values show that the ON $\rightarrow$ OFF structural transition is more costly than the OFF $\rightarrow$ ON one for both the wildtype and Q61L. However, both transitions have higher cost in the Q61L variant, indicating a significant change of the energy landscape upon this mutation.

Table 6.1: The lowest-cost paths and the expected number of transitions are shown for the structural transitions between the ON and OFF states in both the wildtype and Q61L variants.

Sequence	Transition	Min Cost	Exp. Nr. Trans
WT	OFF $\rightarrow$ ON	12.9	3.4×10^{8}
	ON $\rightarrow$ OFF	16.5	3.9×10^{10}
Q61L	OFF $\rightarrow$ ON	20.9	1.9×10^{12}
	ON $\rightarrow$ OFF	24.3	3.8×10^{14}

The lowest-cost paths for each of these transitions in the wildtype are shown in Figure 6.2(a). For ease of visualization, the paths are mapped onto the top two PCs. The color scheme follows the energy variance. Figure 6.2(a) shows that both structural transitions go over an energy barrier, as also reflected in the costs shown for the wildtype sequence in Table 6.1. Moreover, the ON → OFF transition spends more time getting out of a deeper and wider ON basin onto the OFF basin.



(a) (b)

Figure 6.2: The left panel shows the minimum cost paths (in terms of energy) for the wildtype sequence between the OFF and ON states. This plot is rendered in the PCA space created by our EA algorithm for sampling. The right panel shows the energetic profile of the lowest-cost paths when transitions from the ON state to the OFF state for the wild type and Q61L mutant sequences.

The detailed energetic profiles of the lowest-cost paths for the ON → OFF transition in the wildtype and Q61L variant are shown in Figure 6.2(b). The Rosetta all-atom energy is shown for each vertex in these paths, but the path lengths are normalized to allow a direct comparison between the two sequences. Figure 6.2(b) clearly shows that the Q61L mutation magnifies the energy barrier that Ras has to cross in the ON → OFF structural transition. These results are in qualitative agreement with other studies [187] and allow concluding that the transition from the ON to the OFF state is made substantially more

difficult upon the Q61L mutation in Ras. It is important to note that the mutation does not affect the stability of the ON and OFF structural states, since the potential energies of the corresponding states remain the same between the wildtype and variant.

Finally, the first-step analysis is applied to measure and compare the expected number of transitions in each setting. These results are related in column 4 in Table 6.1. Comparison of these results for the wildtype sequence shows that the expected number of transitions to allow switching from the ON to the OFF state is two orders of magnitude higher than from the OFF to the ON state. This also holds for the Q61L variant, though switching from ON to OFF and vice versa becomes more difficult in the variant than in the wildtype.

Taken altogether, these results suggest that a careful realization of the SRS framework may allow a more detailed understanding of the role of sequence mutations in misfunction. In our particular application to the wildtype and Q61L variant of Ras, the results support the hypothesis that the Q61L mutation does not remove the ON and OFF basins from the energy landscape but instead slows down the switching of Ras between these states.

6.4 Conclusions

This chapter has proposed an efficient algorithmic realization of the SRS framework to model structural transitions in dynamic proteins that are known to be conformational switchers and are involved in proteinopathies. Application on sequence variants of Ras shows promising results. The realization we pursue here benefits from dense sampling of the search space of interest, which is generally hard to obtain. In the next chapter, we investigate this issue further by focusing on sampling techniques that are integrated with the connectivity stage. Moreover, we investigate various issues regarding the balance between computational efficiency and multitude of sampled paths, particularly when actual local planners are integrated to realize edges rather than estimate their feasibility. It is worth emphasizing that the work in this chapter has provided insight into how important information can be obtained by exploiting analogies between the notion of a roadmap and that of an MSM. One can anticipate that further MSM-based analysis and calculations may be pursued to

enrich the predictive power and detail obtained by such methods for modeling structural transitions in protein systems.

Chapter 7: SPIRAL – A Roadmap Based Method for Protein Motion Prediction

In chapter 6 we identified two main issues with adapting a roadmap-based approach to the problem of modeling structural transitions in protein molecules. In particular, we identified sampling as a critical component to obtain a dense representation of regions of the conformational space likely to contribute to successful queries, and steering as critical to compute motions between two nearest-neighbor conformations, to then obtain a detailed motion path.

To address sampling in chapter 6, we made use of a stochastic optimization procedure that was highly specific to the protein molecule under investigation. The procedure identified the relevant search space and its dimensionality a priori, based on dimensionality reduction of available stable and semi-stable structures of wildtype and variants of the protein. An evolutionary algorithm was then devised to exploit this search space and populate it with local minima conformations. This procedure, while highly effective, is not general and cannot be applied to any protein molecule. For many proteins, we do not have a rich collection of diverse experimentally-obtained structures in order to define the search space of interest. For others, even if such structures exist in structure databases, there is no guarantee that linear dimensionality reduction, which allows directly obtaining samples in the lower-dimensionality embedding, will be effective. In fact, predominantly, conformational spaces of complex dynamic proteins are shown to be nonlinear [188, 189].

In this chapter, we pursue a sampling procedure that is generally applicable to any protein molecule. It should be noted that this procedure is not expected to provide better sampling than highly specific ones, tailored to a protein under investigation, as is often the case with tradeoffs between general and highly specific algorithms. However, our goal is to

have a general procedure that applies to proteins of different sizes and is yet sensitive to the distance that needs to be traveled over paths connecting a start to a goal structure.

We investigate two main ideas in this direction, fully realizing that dense sampling in regions of interest is an outstanding challenge for protein conformational spaces. The first idea is to make use of a set of (perturbation) operators that employ moves of different granularities. In previous work, we have predominantly employed molecular fragment replacements as moves. With such moves, we can control fragment length as a way to alternate between large and small jumps in conformational space. However, such moves do not provide as much granularity as single dihedral angle replacements. The latter provides more granularity but cannot be used to rapidly generate diverse conformations covering the conformational space. For this reason, we introduce a set of perturbation operators that use moves of different granularities. We introduce a probabilistic scheme that at each iteration selects an operator to be employed for populating the conformational space with low-energy conformations. We note that the idea of employing different types of operators bears some similarities to related efforts in robot motion planning, where different random sampling strategies are considered and switched over through a probabilistic scheme [190]. The second idea we investigate is to focus the sampling to regions of conformational space nearby the start and goal structures provided to the query. We do so by building over the tree-based work we presented in chapter 5. We essentially provide boundaries for the sampling procedure and further guide sampling to populate levels of a progress coordinate in an effort to build a discrete representation of regions likely to contribute to a successful query.

In chapter 6 we circumvented the issue of steering by employing probabilistic edges to connect two nearby conformations. The intuition behind this strategy was that no local planner was necessary to realize an edge; if two conformations were nearby and their energies passed the Metropolis criterion, we essentially could assume that the edge could be realized through low-level motions. Generally, one cannot make this assumption, particularly when sampling is not guaranteed to be uniformly dense. This is likely to be the case even when

employing a powerful sampling strategy and applying to it ever-increasing protein chains. In the absence of domain-specific/protein-specific components, dense sampling cannot be expected on the conformational space of a protein of 300 amino acids or more, even when providing boundaries; the latter can span anywhere from 3Å to $>$ 13Å distance between start and goal structures. The result of non-uniformly dense sampling is that edges spanning larger distances may need to be allowed in order to have a connected graph or a connected component containing the query structures.

In this chapter we have to address the steering issue by pursuing complex local planners that essentially solve the same motion computation problem but for conformations that are closer to each-other than the query structures. The task of these planners is to actually realize edges by providing a series or trajectory of conformations with enough resolution (sufficiently small distance between adjacent conformations in the trajectory). We note that, typically, in baseline implementations of PRM and even in existing adaptations for protein folding/unfolding, ligand binding, and motion computation, local planners are straight-line planners. These planners conduct straight-line interpolations between two conformations over the DOFs, typically backbone dihedral angles, to obtain intermediate conformations. The resulting conformations are evaluated in terms of energy. In some implementations, if any intermediate conformation has energy above some predetermined, arbitrary threshold, the conformation and the entire edge is rejected and considered infeasible. In other implementations, the rejection is probabilistic, employing the Metropolis criterion to determine whether the protein can transition between two consecutive conformations in the straight line. Such planners are rather simplistic. They have a high probability of producing conformations that are infeasible, particularly when an edge is placed between conformations that are not nearby in conformational space. These planners have been demonstrated be deficient even for robot motion planning problems in the manipulation domain, where sampling is also difficult and edges may span different distances in the search space [191].

Therefore, the direction pursued in this chapter is to employ complex, probabilistic path planners, that are able to address essentially simplified versions of the motion computation

problem. However, such planners may be computationally expensive, particularly when asked to realize a difficult edge. Information on difficulty is not known a priori, and probabilistic path planners are not complete. They are probabilistically complete, at best; that is, if a solution does not exist, they can run indefinitely. In the limit, as computation time goes to infinity, if a solution exists, they will be able to find it, but this may not be in practical computational time. For this reason, the approach we pursue here is to first have a running estimate of an edge's difficultly based on the time a planner has spent on realizing it and to place an upper bound on the time a planner has to realize an edge. These two ideas essentially make the case for having a two-level approach to our motion computation problem, a PRM over probabilistic path planners. The latter can be tree-based, roadmap-based, or other.

In this chapter we pursue such a two-level approach. This approach has been originally introduced in [191] as the fuzzy PRM method. In the original introduction for robot manipulation planning, fuzzy PRM was conceptualized as a PRM over PRMs. That is, the global planner built a roadmap of (lazy) pseudo-edges, and the local planners were assigned time to realize selected pseudo-edges. To make good use of resources, at every iteration, a promising path would be identified, with remaining unrealized edges, and local planners would be assigned to work on such edges until a predetermined time expired. At the end, difficulty estimates of remaining unrealized edges were updated in order to then direct the local planners to other possibly more promising paths in the second iteration.

In this chapter we build over the fuzzy PRM approach, but we introduce specific algorithmic components to address sampling for protein conformational spaces and diverse probabilistic planners to address steering and realization of edges in possibly very sparsely-sampled regions of the conformational space. We extend the treatment to obtain more than one path (the original fuzzy PRM method in [191] stopped as soon as a path was realized), so that we can sample diverse paths according to essentially an implicit prioritization scheme. We further adapt the method to make it applicable to rather high-dimensional problems that we are forced to address for proteins; for instance, the sampling procedure

is not detached as in the original fuzzy PRM and PRM formulations. We augment the roadmap with more conformations on regions determined difficult by local planners during the connectivity phase. This is critical to address the non-uniformity of sampling for the complex conformational spaces we address here. Finally, we extend the treatment to queries beyond a specific start-goal pair. Instead, to allow application on proteins with possibly more than two known functional structures, we introduce the notion of ℓ landmarks to keep track of functional structures of a protein. Sampling is guided by the presence of such structures, and paths are sampled in order of difficulty to solve any of the $\ell!$ queries. The motivation for this more general setup is to allow obtaining maximal information from one roadmap.

Specifically, the more general problem we address in this chapter is the following: ℓ landmark structures are provided as input. The sought output is an ensemble of valid paths connecting any pair of landmarks, sampled within a user-determined computational time limit. Path validity, as in the binary setup in the two previous chapters, considers the energetic credibility of the path. In addition, in this chapter, validity also considers resolution constraints (distance between adjacent conformations in the path).

Algorithm 7.1 The *SPIRAL* framework for model structural transitions in proteins

---

Input: $P_1, P_2, ..., P_\ell$ ▷ P is a set of functional states

Output: $\Omega = \left\{\Pi_{ij}^{(1)}, \Pi_{ij}^{(2)}, ..., \Pi_{ij}^{(n)}\right\}$ ▷ ensemble of paths

$\forall p \in \Omega$

- $\Pi_{ij}^{(p)} = C_{ij,0}, ..., C_{ij,t}, ..., C_{ij,\tau}$
  - $0 \leq t \leq \tau$
  - $C_i \in P$, $C_j \in P$, $i \neq j$
  - $C_i = C_{ij,0}$; $C_j = C_{ij,\tau}$
  - $Valid(\Pi_{ij}^{(p)}) = True$

From now on we refer to our framework as *SPIRAL* for the Stochastic Protein motIon

Roadmap ALgorithm. It is worth noting that this is a framework, and different algorithmic components can be pursued by later researchers to investigate different algorithmic realizations. What we describe and analyze in this chapter is a first attempt to essentially provide a roadmap for researchers interested in pursuing protein motion computation with a robotics-inspired path sampling approach.

7.1 Methods

We first provide an overview of the main algorithmic components of *SPIRAL* and then describe each one of them in detail.

7.1.1 Main Components of *SPIRAL*

SPIRAL consists of 3 stages, sampling, connectivity building, and analysis. The sampling stage generates an ensemble of samples, Ω, that provide a discrete representation of the conformational space. The connectivity building stage builds a graph or roadmap $G = (V, E)$ over Ω. The vertex set is populated with conformations in Ω, and pseudo-edges joining neighboring conformations are then added to the edge set using techniques detailed below.

As described above, *SPIRAL* implements a two-level approach. The pseudo-edges added to the roadmap are not checked for energetic feasibility. In effect, this is a lazy scheme, introduced originally in Lazy PRM [183] for robot motion planning and then extended in fuzzy PRM [191] to control the computational cost of the connectivity building stage by realizing selected pseudo-edges. When probabilistic local planners are employed, they can consume computational resources attempting to realize a pseudo-edge that may not be possible. For this reason, only pseudo-edges are added in the first level. Pseudo-edges are assigned a weight to reflect their estimated difficulty of being realizable. At initialization, all pseudo-edges are determined equally difficult. A query is then performed, and the lowest-cost path, using the assigned pseudo-edge weights, is reported and pushed for checking to the second level. This level assigns all yet-to-be-realized pseudo-edges in the path to

probabilistic local planners. Each planner is assigned a computational budget, time T, to realize a pseudo-edge. If a planner fails to realize an edge within the allocated budget, the probability that the pseudo-edge is realizable (and hence, the weight that the first level sees for that pseudo-edge increases) is downgraded according to a heuristic function that takes into account the cumulative time spent on that pseudo-edge. This information is passed to the first level, which starts the process anew, querying the roadmap for the next lowest-cost path.

In this iterative interplay between the first and second levels, over time, the pseudo-edges that are the most difficult to realize will be assigned high weights and will thus be unlikely to participate in the lowest-cost path pushed by the first level to the planners in the second level. This dynamic interplay apportions the computational resources in a manner that promotes rapid discovery of a path connecting a start to a goal vertex. In [191], as soon as all pseudo-edges in a path are realized, the algorithm terminates. In our adaptation the goal is to sample multiple paths, thus the process continues until a requested number of paths are obtained or a total computational budget has expired. We also make use of the K shortest-path algorithm [184] to report K shortest paths connecting ℓ landmark structures.

In the final stage, once *SPIRAL* has obtained $\leq k$ lowest-cost paths, the focus shifts to comparing paths based not on pseudo-edge weight estimates but instead on energetic feasibility. Pseudo-edges now are replaced with the actual ones constructed by the local planners. New conformations sampled by the local planners to realize pseudo-edges are added to the vertex set of the roadmap. Edge weights are now based on the Metropolis criterion, and the resulting graph is queried for lowest-cost paths. Various types of analysis can then be conducted over these paths, whether in terms of energetic profile or proximity to landmark structures.

We now proceed to relate details under each of the three stages of *SPIRAL*.

7.2 Sampling

The objective of the sampling stage is to obtain an ensemble of conformations Ω that will constitute the vertex set of the roadmap G. This stage consists of a cycle of selection and perturbation operators. A selection operator selects a conformation within the current sampled ensemble. Once selected, a perturbation operator is then sampled from a set of available ones and applied to the selected conformation to generate a new one. The generated conformation is checked for energetic feasibility prior to addition to the ensemble Ω. The process repeats until $|\Omega|$ reaches a pre-determined value. The pseudo-code for the sampling stage is shown in Algorithm 7.2.

7.2.1 Selection Operator

In this setting, we build over the selection procedure originally introduced in FeLTr and modified in our tree-based motion computation algorithm in chapter 5. A progress coordinate, ΔR, can be defined for each conformation C_i and a specific start-goal structure pair (C_s, C_g) in the set of ℓ landmark structures P, as in: $\Delta R = \text{lRMSD}(C_s, C_i) - \text{lRMSD}(C_g, C_i)$.

The ΔR coordinate is used to guide sampling towards under-sampled regions in the conformational space. For each pair of landmark structures (C_s, C_g), a 1d grid is defined over the range $[-\text{lRMSD}(C_s, C_g) - 2, \text{lRMSD}(C_s, C_g) + 2]$. Each cell in the grid is 1Å wide. All conformations in Ω are projected onto this grid. To bias the selection of conformations from under-explored regions of the conformational space, a weight w_c is associated with each cell in a given grid, as in: $w_c = \frac{1}{(1+nsels)*nconfs}$, where $nsels$ is the number of times the cell has been selected in the sampling procedure, and $nconfs$ is the number of conformations projected onto that cell. In this way, each conformation in the growing ensemble Ω has $\binom{\ell}{2}$ projections, one in each of the $\binom{\ell}{2}$ grids that keep track of how the conformational space is covered with respect to the progress coordinate.

SPIRAL's selection operator, shown on lines 18-19 in Algorithm 7.2, proceeds as follows. First a pair of landmark structures is selected uniformly at random among the $\binom{\ell}{2}$ pairs. The

Algorithm 7.2 The algorithm for the sampling phase of *SPIRAL*.

```
1: procedure PERTURB(C,PerturbOps,MaxAttempts)
Input:
    C                                          ▷ conformation to perturb
    PerturbOps                 ▷ Set of perturbation operators and constraints
    MaxAttempts                              ▷ Maximum perturbation attempts
Output: CNew                                   ▷ perturbed conformation
2:      T = RetrieveTemperature(C)
3:      CNew ← NULL
4:      while attempts < maxAttempts do
5:          POperator = SelectPerturbOps(PerturbOps)
6:          CNew = POperator.Apply(C)
7:          if POperator.Validate(CNew,T,Ω) then
8:              break;
9:          end if;
10:     end while
11:     return(CNew);
12: end procedure
```

```
Input:
    P_1, P_2, ..., P_ℓ                          ▷ Set of functional states
Output:
    Ω                                              ▷ ensemble of states
13: for ∀p ∈ P do
14:     Ω = Ω ∪ p
15:     UpdateProjection(p)
16: end for
17: while |Ω| < SamplesRequested do
18:     LP = SelectLandmarkPair()                  ▷ Select pair of structures
19:     C = SelectExistingSample(LP)   ▷ Select sampled based on 1d grid for LP
20:     CNew = Perturb(C)                          ▷ Create new sample
21:     if CNew is valid then
22:         Ω = Ω ∪ CNew                           ▷ Append to ensemble
23:         UpdateProjection(CNew)
24:     end if
25:     UpdateReactiveTemp(C,CNew)      ▷ Update statistics and temperature
26: end while
```

selection of the pair then determines the 1d grid to be employed. A cell is sampled from the selected grid using a probability distribution function defined over the weights associated with grid cells as above. Once a cell within the 1d grid is selected, a conformation from that cell is then selected uniformly at random.

7.2.2 Perturbation Operators

SPIRAL employs a set of perturbation operators in order to make moves of different granularities in conformational space in the sampling stage. Each perturbation operator has to satisfy a set of constraints. One of the constraints enforces energetic feasibility of generated conformations. The energy value of a conformation $c^{'}$ generated from a selected conformation C, measured through the Rosetta energy function, is compared to the energy value of C through the Metropolis criterion. If this fails, the conformation $C^{'}$ is not added to the ensemble. If it passes, $C^{'}$ is checked for satisfaction of distance-based constraints. Additional constraints are introduced on the minimum lRMSD ϵ_{min} of $C^{'}$ to any other conformation in the ensemble Ω and the maximum lRMSD δ of $C^{'}$ to the ℓ landmark structures. The first constraint prevents redundant conformations from being added to Ω. The second constraint prevents sampling from veering off in regions of the conformational space deemed too far from the landmark structures to be useful for participating in paths connecting these structures. While ϵ_{min} is a parameter taking values anywhere from $0.5-2$Å and tuned on the specific system under investigation, a reasonable value for δ is 150% of the maximum lRMSD between any pairs of landmark structures.

The idea behind making various perturbation operators available to *SPIRAL* is to allow *SPIRAL* to select the perturbation operator deemed most effective based on features of the conformational space and the specific problem. For instance, when the goal is to connect landmark structures that reside far way from one another in conformational space, a perturbation operator capable of making large jumps in conformational space is first desirable. Afterwards, to be able to make connections between such conformations, other perturbation operators capable of making smaller jumps may be more effective. We consider here three

perturbation operators, detailed below. An optimal weighting scheme that is responsive to emerging features of the search space is difficult to formulate and beyond the scope of the work here. However, we have been able to empirically determine a weighting scheme that is effective on most protein systems with landmark structures of various pairwise lRMSD values. We now proceed to describe each of the perturbation operators in detail.

Molecular Fragment Replacement Operator

The molecular fragment replacement technique has been described in section 4.1.1. Here, we employ two different fragment lengths, 9 and 3. Note that *SPIRAL* allows for any fragment length to be used, but we determine that these two fragment lengths can be used to balance between large jumps (fragment length 9) and small jumps (fragment length 3).

Single Dihedral Replacement Opreator

This perturbation operator modifies a single backbone dihedral angle at a time in order to allow making very small moves in conformational space. Given a selected backbone dihedral angle in a selected conformation, a new value from it is obtained using a normal distribution $\mathcal{N}(\mu, \sigma)$. Normally, the angle to perturb is selected uniformly at random. This operator offers the option of biasing the selection of dihedral angles to promote selection of those that differ most between a selected conformation and a landmark structure. Essentially, each angle is weighted by its absolute difference between a selected conformation, C, and the selected landmark structure (line 18 of Algorithm 7.2).

Conjugate Peak Refinement Operator

The final perturbation operator *SPIRAL* employs is an adaptation of the conjugate peak refinement (CPR) method originally introduced in [192]. Briefly, CPR produces a series of intermediate conformations to approximate a potential reaction path between two given (start and goal) conformations p and r. The initial guess of the path is a straight line interpolation between p and r. The highest-energy conformation, x_1, is identified and

subsequently minimized to obtain x_1^*. This results in two path segments, [p x_1^*] and [x_1^* and r]. The process of identifying (and then minimizing) the highest-energy conformation over the existing path segments continues until a desired resolution and energy profile are obtained.

In its original form, CPR requires an energy function that is continuous and for which the first derivative can thus be defined. Here we do not have direct access to the first derivative of the Rosetta suite of energy functions. As an alternative, we employ Rosetta's *relax* function, which performs a simulated-annealing minimization after adding side chains to backbone-resolution conformations. In our employment of *relax*, we constrain the movement of backbone atoms so the minimized conformation x_i^* remains close to x_i. The pseudo-code for CPR is shown in Algorithm 7.3 and is illustrated in Figure 7.1.

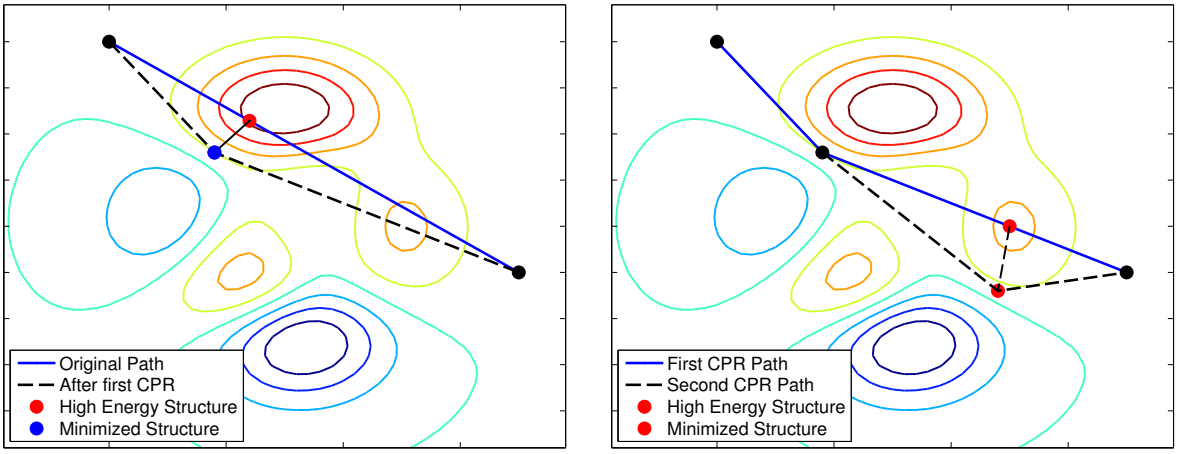
Algorithm 7.3 The Conjugate Peak Refinement algorithm [192]

Input:
Function states C_s, C_t
ϵ, interpolation interval
Output: Path $C_s, C_1, C_2, ..., C_n, C_t$

```
P ← InterpolateInitialPath(ε)
while Time AND EnergyOK = FALSE do
    H ← SelectHighEnergy(P)
    H_Min ← Miniimize(H)
    P ← SegmentPath(P,H,H_Min,ε)
end while
```

We make use of CPR as follows. First, two conformations are selected from Ω using our selection operator. CPR is then applied, as modified above, to produce intermediate conformations. Intermediate conformations that pass the energetic and distance constraints detailed above are added to the ensemble Ω. This CPR-based perturbation operator is shown in pseudo-code in Algorithm 7.4.

Figure 7.1: A cartoon example of the CPR algorithm. The left side shows the initial interpolated path in blue, with the highest energy conformation shown in red. This structure undergoes an energy minimization, resulting in the blue point. A new path is now constructed via the blue point. The right panel illustrates the next iteration of the algorithm



(a) (b)

Algorithm 7.4 CPR as a perturbation operator

Input: Function states C_s, C_t, Ω
Output: Path $C_s, C_1, C_2, ..., C_n, C_t$

```
P ← InterpolateInitialPath(R)
for p ∈ P do
    p_min ← RosettaRelax(P)
    if Validate(p_min) then
        Ω ← Ω ∪ p_min
    end if
end for
```

7.2.3 Reactive Temperature Scheme

We have found that the combination of energetic and distance constraints make it increasingly difficult to obtain constraints-satisfying conformations as the ensemble Ω grows. Therefore, we tune the energetic constraint by controlling the effective temperature used in the Metropolis criterion through a reactive temperature scheme similar to the one employed by the tree-based motion computation algorithm introduced in chapter 5. We maintain a temperature value T_c for each cell c of the 1d grids over the progress coordinate. Each cell's

temperature is adjusted every s steps (typical value employed is 25). The temperature of a cell, T_c, is increased if the last s selections of that cell have resulted in no conformations being added to Ω. If conformations are added to Ω more than 60% of the time within a window of s steps, T_c is decreased. Increases and decreases occur over adjacent temperature levels per the proportional scheme detailed in section 5.2.5.

7.3 Connectivity Building

The connectivity building stage starts by adding all conformations in Ω to the vertex set. Pseudo-edges are then identified and weighted. The rest of this stage then consists of the interplay between path identification and path realization. The graph is augmented with more conformations as local planners identify difficult regions. The interplay continues until K lowest-cost paths are determined or the computational budget has been exhausted.

7.3.1 Identification and Weighting of Pseudo-edges in the Roadmap

This process is shown in pseudo-code in Algorithm 7.5. For each conformation/vertex $v \in V$, its k nearest-neighbors are identified. For each identified neighbor, directional pseudo-edges are added with v. Additional pseudo-edges are added by identifying any vertex $< \epsilon_{max}$ from v that lies in a different connected component from v. Typical values for $k = 10$ and $\epsilon_{max} = 5\mathring{A}$. All added pseudo-edges are assigned an initial weight of value 1.

7.3.2 Path Query and Path Realization Interplay

The pseudo-code for this process is shown in Algorithm 7.6. A pair of landmark structures are selected uniformly at random over the $\ell!$ permutations. The roadmap is then queried for the lowest-cost path. We utilize Yen's K-Shortest path algorithm to identify the lowest non-zero cost path in the graph and allow us to continue obtaining paths after the first path has been successfully realized.

Given an identified path, a local planner is assigned to any of the unrealized edges (the

Algorithm 7.5 Roadmap construction pseudo-code.

Input:
Ω ▷ The ensemble generated during the sampling stage
k ▷ number of nearest neighbors
K ▷ number of neighbors in different connected components
ϵ_{max} ▷ maximum connect distance for K
Output: $G = (V, E)$ ▷ the graph that encodes the roadmap

```
V = Ω
for ∀c ∈ Ω do
    Neighbors = NearestNeighbors(c,k)
    for ∀t ∈ Neighbors do
        E = E ∪ e(v, t)
    end for
    NeighborsCC = NearestNeighborsCC(c,K,ε_max)
    for ∀t ∈ Neighbors do
        E = E ∪ e(v, t)
    end for
end for
```

local planner is described below in section 7.3.3). The planner is given a fixed computational budget, time T. If the local planner succeeds, the pseudo-edge it has realize is assigned a weight of 0 to indicate the pseudo-edge is resolved. If the local planner fails, the pseudo-edge is reweighted as shown in Equation 7.1.

$$E_{score} = 0.7 \cdot \text{CallsToPlanner} + 0.3 \cdot (\text{ClosestNode} - \text{RequireResolution})^2 \tag{7.1}$$

CallsToPlanner tracks the number of times the planner has been requested to work on a particular pseudo-edge, ClosestNode is the node in the tree constructed by the local planner that is closest to the vertex v in the directed pseudo-edge (u, v). For the planner to be successful, it must also generate a path that is within RequireResolution lRMSD of the vertex v, so this value is also employed in Equation 7.1.

An additional feature of *SPIRAL* is its ability to learn from failures. When a local planner has failed to complete a path more than RefineLimit times, *SPIRAL* augments the graph with conformations identified by the local planner that are otherwise invisible to the top layer. We now proceed to relate details on the local planner and the augmentaion

Algorithm 7.6 The algorithm for the connectivity phase of *SPIRAL*.

Input: $G = (V, E)$ ▷ The roadmap encoded as a graph

```
for RefineCount < RefinementMax do
    LP = SelectLandmarkPair()                ▷ Select which pair of structures to refine
    LP_path = ComputeLowCostPath(LP)
    for ∀ segment ∈ LP_path do
        if segment.score != 0.0 then
            RefineSegment(segment)
            if segment.score != 0.0 AND segment.refineCount mod RefineLimit = 0 then
                AugmentRoadmap(segment);
                segment.refinementCount = 0
            end if
        end if
    end for
    UpdateLPStats()                          ▷ Update Stats on paths per LP pairing
end for
```

procedure.

7.3.3 Local Planner

The local planner employed by *SPIRAL* is an adaption of the tree-based planner we investigated in Chapter 5. The adaptation consists of diversifying the types of perturbation operators employed in the expansion of the tree. In addition to the molecular fragment replacement technique with a fragment length of 3, as in Chapter 5, we also employ two additional operators, gausssian sampling and biased gaussian sampling. The latter selects more frequently dihedral angles with higher differences between the current conformation and goal conformation. A probabilistic scheme is designed to select each of these three operators during each expansion of the tree in the local planner. The particular scheme employed is reported in Results. While not fine-tuned, the scheme assigns higher probability of selection to operators capable of making larger moves when the distance that needs to be bridged to reach the goal vertex/conformation is large.

7.3.4 Augmenting the Graph

Some regions of conformational space may present significant challenges for *SPIRAL* to connect through local planners. This can be due to high energetic barriers that exist or because of inadequate sampling. To address this issue, *SPIRAL* makes use of a feedback mechanism to augment the graph by adding samples in these regions, as determined by the local planners. When a local planner encounters difficulty realizing a pseudo-edge more than $RefineLimit$ times (a typical value for this parameter is 25), the graph is augmented with conformations produced by perturbation operators. These operators include not only the molecular fragment replacement technique and the biased and unbiased gaussian samplers, but also CPR. The augmentation is shown in pseudo-code in Algorithm 7.7. The particular weights assigned to each operator within the probabilistic scheme are shown in Results.

Algorithm 7.7 The algorithm for the augmenting the graph during the Connectivity stage.

Input:
$e = (u, v)$ ▷ The edge on which the local planner is working
G=(V,E) ▷ The roadmap graph object

```
1: for AugmentCount < AugmentMax do
2:     c = SelectRandomConf(u,v)          ▷ select u or v uniformly at random
3:     POps = SelectPerturbOps()
4:     VNew = {}
5:     CNew = Perturb(C);
6:     if POperator.Validate(CNew,e.T) then
7:         VNew = VNew ∪ CNew
8:     end if
9: end for
10: V = V ∪ VNew
11: for ∀ v ∈ VNew do                     ▷ Connect to roadmap
12:     Neighbors = NearestNeighbors(v,k)
13:     for ∀t ∈ Neighbors do
14:         E = E ∪ e(v,t)
15:         E = E ∪ e(t,v)
16:     end for
17: end for
```

7.4 Analysis

It is worth noting that after it exhausts the computational budget, *SPIRAL* may have yielded $\leq K$ lowest-cost paths. The weights on these paths are not related to energetic feasibility. Therefore, in the final stage, *SPIRAL* reweights the entire graph using the Metropolis criterion as the edge weights. The reweighted graph is then queried for paths, which can be analyzed in terms of energetic profile or distance within which they come of the goal landmark structures.

7.5 Results

SPIRAL is implemented in C++ and experiments are run on the Mason Argo cluster and the Hydra cluster. Three sets of experiments are run for each protein system considered here, depending on the requested size of the sampled ensemble Ω. Three sizes are considered to investigate the scaling in computational time as a function of ensemble size: $|\Omega| \in \{5,000,\ 10,000,\ 20,000\}$. A hard termination criterion is set with regards to the total number of energy evaluations. The sampling stage is terminated if the total number of energy evaluations exceeds $1,000$ times the requested ensemble size. That is, a maximum of 25 attempts are made to obtain a sample. The connectivity building stage is terminated after $10,000$ iterations of the interplay between path query and path realization. This stage may terminate earlier if 250 paths are obtained for all $\ell!$ landmarks as a way to control computational cost. The analysis stage modifies the roadmap as described in Methods and reports the 50 lowest-cost paths. In terms of CPU time, the computational time demands of all these three stages in *SPIRAL* spans anywhere from 2 days for protein systems around 100 amino acids long to 30 days for systems around 700 amino acids long.

7.5.1 Systems of Study

The protein systems we have selected are carefully gathered from published literature in order to provide some comparisons. We note that since motion computation for proteins

is still an emerging research area, not many published methods exist. Moreover, many of them focus on either specific systems or are rather limited by system size. In all, we have been able to collect 7 systems with published results. They are listed in Table 7.1.

Column 1 in Table 7.1 lists the names for these systems, and column 2 shows their lengths in terms of number of amino acids. For most of these systems, two diverse functionally-relevant structures have been extracted from wet-lab literature to serve as start and goal (we consider both directions here) structures. CaM is the only system on which we test *SPIRAL* on its more general setting of $\ell > 2$ landmarks (3 in this case). The final column in Table 7.1 shows the distance between the start and goal structures for each system in terms of lRMSD.

Table 7.1: Protein systems for evaluation.

System	Length	Start ↔ Goal	lRMSD(start, goal)
CVN	101	2ezm ↔ 1l5e	16.01 Å
CaM	140	1cfd ↔ 1cll	10.7 Å
		1cfd ↔ 2f3y	9.9 Å
		1cll ↔ 2f3y	13.44 Å
AdK	214	1ake ↔ 4ake	6.96 Å
LAO	238	1laf ↔ 2lao	4.7 Å
DAP	320	1dap ↔ 3dap	4.3 Å
OMP	370	1omp ↔ 3mbp	3.7 Å
BKA	691	1cb6 ↔ 1bka	6.4 Å

It is worth noting that neither protein size, nor the lRMSD distance between functional states do by themselves define system difficulty. We have observed that the larger systems (in terms of number of amino acids) that exhibit smaller motions (less than 4.5Å lRMSD) between the start and goal structures may require the protein chain to partially unfold before returning to a folded state. The process of unfolding a large, compact structure is computationally costly, as effectively an energy barrier needs to be crossed to get out of the compact state. Indeed, many computational studies avoid computing the motions involved

in transitions from a closed to an open structural state because of this challenge.

7.5.2 Parameter Values

The probabilities with which each of the three perturbation operators are selected by *SPIRAL* during sampling are shown in Table 7.2.

Table 7.2: The perturbation operator set and weights used to select them during *SPIRAL*'s sampling stage.

Perturbation Operator	Probability
Molecular Fragment Replacement (length 3)	0.75
Molecular Fragment Replacement (length 9)	0.20
Gaussian Sampling ($\mu = 0$, $\sigma = 15$)	0.05

As described in Methods, the tree-based planner makes use of molecular fragment replacement and gaussian sampling of dihedral angles during the expansion of the tree. The augmentation stage in *SPIRAL* makes use of these same operators, but also includes CPR. The probabilities associated with these operators by the tree-based planner are shown in Table 7.3. The local planner uses a different scheme depending on the distance between the two conformation/vertices it is asked to connect. While the specific probability distribution is not tuned, the values that we have determined to perform reasonably essentially promote operators that are capable of making smaller moves when the requested distance to be bridged is < 2.5Å; in contrast, for larger distances, the operators that make larger moves are given higher probability of selection. The particular threshold of 2.5 used here to switch the probabilistic scheme is based on an earlier finding from our work on tree-based motion computation (chapter 5). In that work we showed that molecular fragment replacement can result in step sizes greater than 2.5Å. During graph augmentation, we introduce CPR in order to explore the space surrounding the start and goal conformations provided to the local planner.

Table 7.3: The perturbation operator set and weights used to select them during *SPIRAL*'s connectivity building stage.

Connectivity Building	Perturbation Operator	Probability
Local Planner ($>$ 2.5 Å lRMSD)	Molecular Fragment Replacement (length 3)	0.70
	Gaussian Sampling ($\mu = 0$, $\sigma = 15$)	0.15
	Gaussian Sampling (biased) ($\mu = 0$, $\sigma = 15$)	0.15
Local Planner ($\leq$ 2.5 Å lRMSD)	Molecular Fragment Replacement (length 3)	0.20
	Gaussian Sampling ($\mu = 0$, $\sigma = 15$)	0.40
	Gaussian Sampling (biased) ($\mu = 0$, $\sigma = 15$)	0.40
Augmentation	Molecular Fragment Replacement(length 3)	0.20
	Gaussian Sampling ($\mu = 0$, $\sigma = 15$)	0.40
	Gaussian Sampling (biased) ($\mu = 0$, $\sigma = 15$)	0.40
	CPR	0.05

The ϵ_{min} parameter is chosen specifically for each system. When the distance between start and goal structures is $\leq$ 4.5Å, we designate the distance as small, and investigate three settings for ϵ_{min}, 0.5, 0.75, and 1.0. When the distance is $>$ 4.5 but $\leq$ 6.0Å, we designate the distance as medium, and investigate three settings for ϵ_{min}, 0.75, 1.0 and 1.5. Distances $>$ 6Å are designated as large, and for the corresponding systems we investigate three settings for ϵ_{min}, 1.0, 1.5, and 2.0. These are listed in Table 7.4.

Table 7.4: Values investigated for ϵ_{min} for each protein system.

System	Distance Designation	Values for ϵ_{min}
CVN	Large	$\{1.0, 1.5, 2.0\}$
CaM	Large	$\{1.0, 1.5, 2.0\}$
AdK	Large	$\{1.0, 1.5, 2.0\}$
LAO	Medium	$\{0.75, 1.0, 1.5\}$
DAP	Small	$\{0.5, 0.75, 1.0\}$
OMP	Medium	$\{0.75, 1.0, 1.5\}$
BKA	Large	$\{1.0, 1.5, 2.0\}$

The ϵ_{min} parameter controls how close neighboring conformations will be in the roadmap.

Intuitively, one might believe that smaller ϵ_{min} values would produce a better quality roadmap. Our research indicates that this is not the case. Small values of ϵ_{min} (< 1Å) for systems with distance designations of medium or greater can result in many small cliques being formed in the roadmap around local minima. This is not surprising, particularly for the broad minima that contain the stable and semi-stable landmark structures. On these minima, it is rather easy to sample a very large number of conformations nearby a landmark and thus essentially "get stuck" in the same local minimum. Insisting on a minimum distance separation among sampled conformations forces sampling not to provide refinement or exploitation of a particular local minimum but rather explore the breadth of the conformational space. Not insisting on a minimum distance pushes all the work to obtaining intermediate conformations to bridge local minima to the local planners, which is an ineffective use of computational time.

7.5.3 Systems of Study and Experimental Design

We present three sets of experiments. First, we show the scaling in computational time during the sampling stage as a function of system length and values employed for $\epsilon_{\min}$. In the second and third experiments, we compare paths produced by *SPIRAL* to those obtained by other methods. We focus on $|\Omega| = 10,000$ and $\epsilon_{\min}$ set to the largest of the three values considered (see Table 7.4) for these comparisons, as our investigation indicates that these settings allow the connectivity building stage to realize paths reasonably quickly (data shown below). We first compare the proximity with which paths come to the specified goal structure and then analyze specific paths in terms of their energetic baseline over a baseline method.

7.6 Sampling Stage Analysis

Figure 7.2 shows the CPU time demanded by the sampling stage to obtain an Ω ensemble of $10,000$ conformations for each of the proteins at each of the three distance-dependent $\epsilon_{\min}$ values considered. The times shown represent the average across 3 independent executions.

Figure 7.2 shows that computational demands rise exponentially with respect to protein length. For the same protein, the higher $\epsilon_{\min}$ values also result in higher computational demands, as it becomes harder to find conformations that satisfy the constraints.

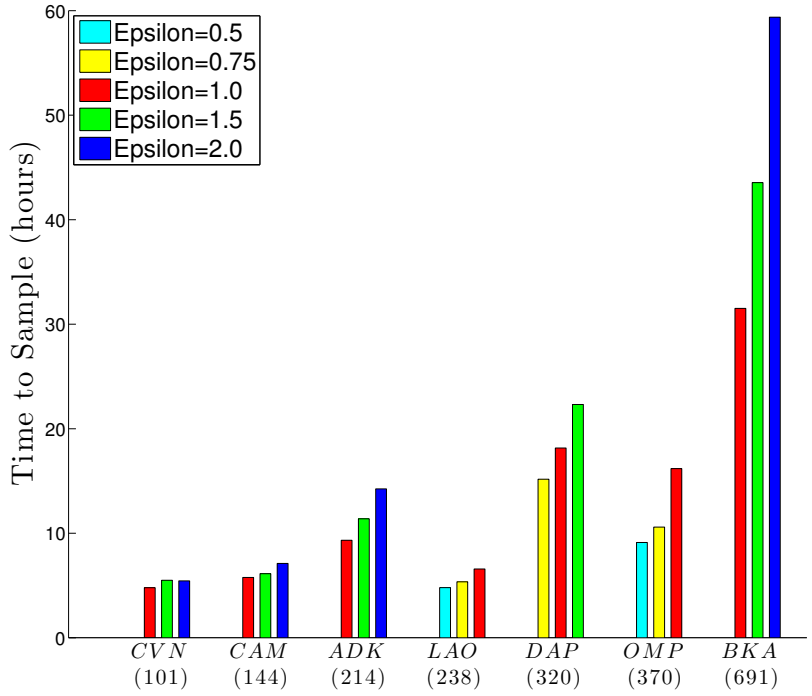


Figure 7.2: CPU time demands of the sampling stage, shown in hours, is an average over three independent executions of *SPIRAL* for each setting considered. For each protein, three settings are considered depending on the $\epsilon_{\min}$ value utilized during sampling.

7.6.1 Analysis of Nearest-Neighbor Calculations

Calculating nearest neighbors in high-dimensional space is a challenging and open problem. Here we have investigated two main techniques for calculating nearest neighbors. In both settings, we have utilized lRMSD as the distance metric. The first technique uses a "brute-force" approach, formulating a full distance matrix and retaining it in memory. The other technique is an adaption of the Geometric Near-neighbor Access Tree (GNAT) [193], which

has been used in other high-dimensional settings as an alternative to kd-trees. In our implementation of *SPIRAL* , the user can select which nearest-neighbor technique should be employed at run-time. The technique is employed very frequently during the sampling stage to enforce the ϵ_{min} constraint. Moreover, *SPIRAL* periodically computes summary statistics on the exploration, including average and percentile statistics on each sample's nearest neighbors. We have found that in such an analysis-intensive setting, the brute-force approach exploiting a distance matrix retained in memory is more computationally efficient than GNAT (data not shown).

7.6.2 Comparison of Paths with Other Methods

We now compare *SPIRAL* to published tree-based methods in [1,57,194,195]. We note that all these methods make use of specific moves. For instance, our own work in [1], summarized in chapter 5, uses molecular fragment replacements of length 3, work in [194] uses an adaptation of RRT with moves consisting of low-frequency modes revealed by normal mode analysis, and work in [57,195] is an adaptation of PDST changing values only for angles that differ between start and goal structures, effectively considering a search space of no more than 30 dimensions. We report here the closest that any path computed by *SPIRAL* comes to the specified goal structure and compare such values on all protein systems to those reported by published work. Columns $4-7$ in Table 7.5 show these values for *SPIRAL* , our tree-based method summarized in chapter 5, and work published by other authors. Column 3 reports some more details on the path with which *SPIRAL* comes closest to the goal structure by listing the maximum lRMSD distance between any two consecutive conformations in the path. *SPIRAL* typically generates paths with conformations closer to the goal structure than other methods.

7.6.3 Comparison of Energetic Profiles

We provide some more detailed results here by relating the energetic profile of the lowest-cost path obtained by *SPIRAL* on two selected systems, AdK and CaM. We compare these

Table 7.5: Column 4 reports the closest distance to the goal structure over all paths obtained by *SPIRAL*. Column 5 shows such distance obtained from our tree-based method summarized in chapter 5 and published in [1]. Columns 6−7 report values obtained by tree-based methods of other authors. Max Step in column 3 refers to the maximum lRMSD distance between any two consecutive conformations in the *SPIRAL* path that comes closest to the goal.

System	Start → Goal	Max Step	Dist to Goal ($\mathring{A}$)			
			SPIRAL	Tree-based [1]	Cortés[194]	Haspel [57, 195]
CVN (101 aa)	2ezm → 1l5e	1.5	1.5	–	2.1	2.1
	1l5e → 2ezm	1.5	1.3	–	–	–
CaM (144 aa)	1cll → 1cfd	3.4	1.46	3.35	–	–
	1cfd → 1cll	2.67	1.12	3.17	–	–
	1cll → 2f3y	2.77	1.26	1.67	–	–
	2f3y → 1cll	3.5	1.12	0.73	–	1.33
	1cfd → 2f3y	3.33	1.26	3.5	–	–
	2f3y → 1cfd	3.48	1.46	3.2	–	–
AdK (214 aa)	1ake → 4ake	3.0	1.86	3.8	2.56	2.2
	4ake → 1ake	3.12	1.33	3.6	1.56	–
Lao (238 aa)	2lao → 1laf	2.0	1.21	–	1.32	–
	1laf → 2lao	3.2	1.90	–	–	–
DAP (320 aa)	1dap → 3dap	1.42	1.5	–	1.31	–
	3dap → 1dap	1.46	0.92	–	–	–
OMP (370 aa)	1omp → 3mbp	1.04	3.04	–	–	–
	3mbp → 1omp	0.91	3.61	–	–	–
BKA (691 aa)	1bka → 1cb6	3.87	1.55	–	2.79	–
	1cb6 → 1bka	3.98	1.69	–	–	–

profiles to those that can be obtained by our adaptation of CPR, as we do not have access to paths obtained via methods published by other authors. For CPR, the resolution distance ϵ is set to 1.0 Å, and 50 cycles of CPR are performed in order to obtain a path. This provides a fair comparison, given that we also analyze 50 paths obtained after the analysis stage in *SPIRAL* and report here the lowest-cost one.

Figure 7.3 shows that on proteins, such as AdK, where the distance between the start and goal structures is large, paths provided by CPR tend to have higher energies than those provided by *SPIRAL*. On systems, such as CaM, where the start-to-goal distance is smaller, CPR can perform comparably to *SPIRAL*. These results illustrate that *SPIRAL* produces good-quality paths, and analysis of these paths can be used to obtain information on protein motions as well as information on possible long-lived intermediate states in dynamic systems.

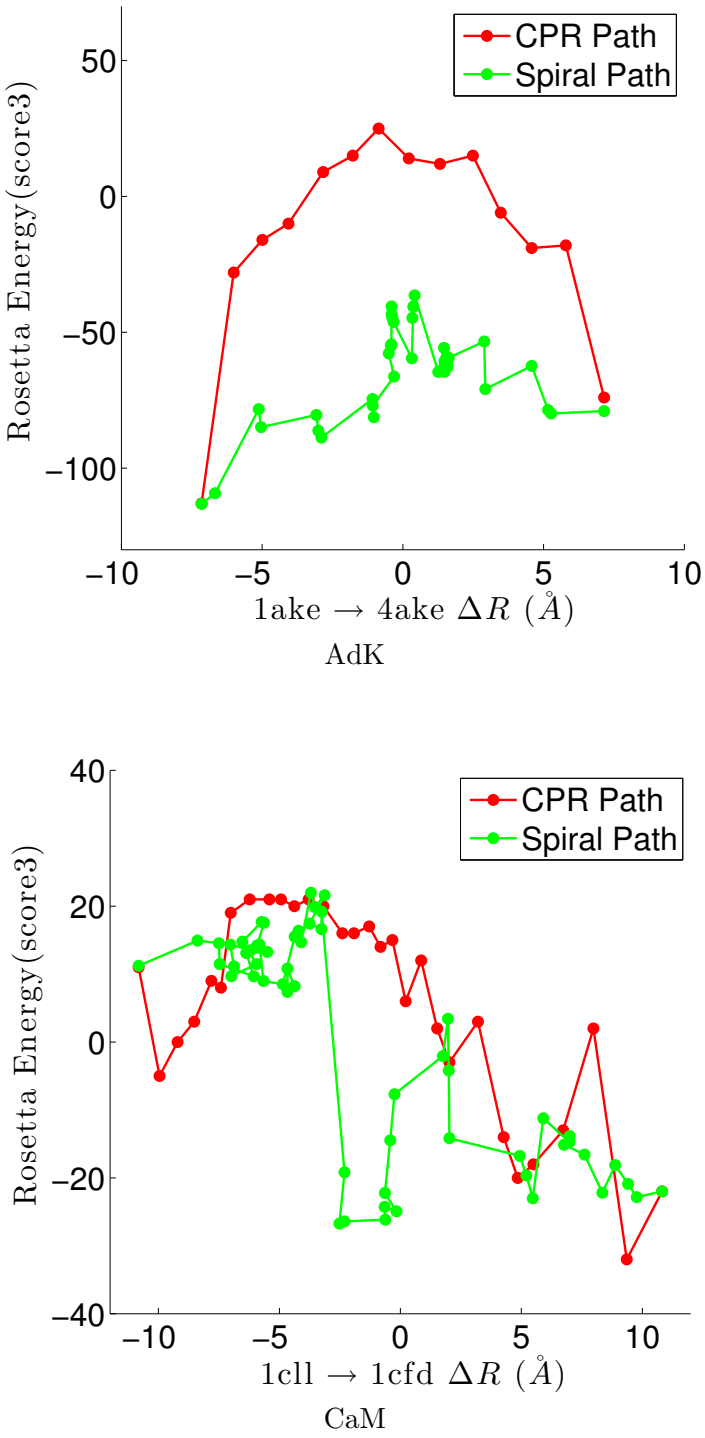


Figure 7.3: Energy profiles of conformational paths computed between 1ake and 4ake of AdK (top) and of CaM (bottom). The red paths are those computed with CPR, and the green ones are computed by *SPIRAL*.

Chapter 8: Conclusions and Future Directions

This thesis has presented novel probabilistic algorithmic frameworks to address three standing challenging in protein modeling research, prediction of function from structure, prediction of the active structure from protein sequence, and mapping of transitions employed by dynamic proteins to switch between stable and semi-stable structures to tune function. The work presented here has advanced the current computational treatment of proteins.

An exploitation of topic-based modeling in machine learning, combined with understanding of protein structure organization, has yielded a novel representation of protein structure that allows efficiently detecting remote protein homologs and, more importantly, automating the process of function annotation for a protein structure. Building over a robotics-inspired optimization framework for adaptive search of the protein conformational space has advanced the problem of decoy sampling and exposed a highly-versatile framework to better understand challenges in *de novo* protein structure prediction. In addition to investigating the impact of various projections in discretization layers employed by the search, our work has shown that a soft energy bias is more effective when pursuing local minima of a distorted energy surface. This is a general result that extends beyond protein modeling research to modeling of complex systems with empirical or semi-empirical cost functions. Finally, analogies with robot motion planning have been pursued in greater detail in this thesis to present novel algorithmic frameworks for the problem of molecular motion computation for the elucidation of structural transitions in proteins.

This thesis advances protein modeling research by extending the size and complexity of systems that can be modeled, as well as the detail and accuracy with which important biological questions on the relationship between protein sequence, structure, dynamics, and function can be answered in silico. For instance, algorithms proposed here to model structural transitions are now able to explain the impact of sequence mutations on protein

function. Our results on Ras and the impact of oncogenic mutations on transitions of Ras between its two main functionally-relevant states are particularly exciting. These results point to the possibility that reliable predictions can be made in silico and that wet and dry laboratory studies may soon complement each-other in understanding and treating disease.

The work presented in this thesis has identified several future directions of interest to possibly diverse communities of researchers in optimization and protein modeling research. The rich algorithmic frameworks presented here consist of various components that can be adapted, modified, and investigated in greater detail depending on the application of interest. While our work under each chapter in this thesis has identified specific future directions on each of the three problems considered here, it is worth reiterating that the work we have described here on a general roadmap-based framework for elucidating structural transitions may present a particularly fertile ground for future research. The two key issues of sampling in the presence of non-trivial and often conflicting constraints and apportioning of computational time in an adaptive manner are themes that have permeated the bulk of the work presented here but become particularly critical and challenging when the objective is to map transitions of a system among various states of interest. In addition, the connection between the stochastic roadmap over a continuous space and a markov state model over discrete states is worthy of further investigation in order to obtain reliable measurements of protein kinetics in silico in a reasonable amount of time.

While the application domain of the computational research presented in this thesis is protein modeling, the algorithmic techniques proposed here are of general utility to other domains in computer science. The research proposed here may benefit other domains that pursue effective optimization for complex systems with continuous and discrete variables, where variables number in the hundreds or more, and impose non-trivial implicit constraint on one another.

Bibliography

Bibliography

[1] K. Molloy and A. Shehu, "Elucidating the ensemble of functionally-relevant transitions in protein systems with a robotics-inspired method," *BMC Struct Biol*, vol. 13, no. Suppl 1, p. S8, 2013.

[2] W. Humphrey, A. Dalke, and K. Schulten, "VMD - Visual Molecular Dynamics," *J. Mol. Graph. Model.*, vol. 14, no. 1, pp. 33–38, 1996, http://www.ks.uiuc.edu/Research/vmd/.

[3] R. Kolodny, P. Koehl, and M. Levitt, "Comprehensive evaluation of protein structure alignment methods: Scoring by geometric measures," *J. Mol. Biol.*, vol. 346, p. 11731188, 2005.

[4] S. Shivashankar, S. Srivathsan, B. Ravindran, and A. V. Tendulkar, "Multi-view methods for protein structure comparison using Latent Dirichlet Allocation," *Bioinformatics*, vol. 27, pp. i61–i68, 2011.

[5] A. Shehu, "An ab-initio tree-based exploration to enhance sampling of low-energy protein conformations," in *Robot: Sci. and Sys.*, Seattle, WA, USA, 2009, pp. 241–248.

[6] A. R. Fersht, *Structure and Mechanism in Protein Science. A Guide to Enzyme Catalysis and Protein Folding*, 3rd ed. New York, NY: W. H. Freeman and Co., 1999.

[7] C. Soto, "Unfolding the role of protein misfolding in neurodegenerative diseases," *Nat Rev Neurosci*, vol. 4, no. 1, pp. 49–60, 2003.

[8] ——, "Protein misfolding and neurodegeneration," *JAMA Neurology*, vol. 65, no. 2, pp. 184–189, 2008.

[9] V. N. Uversky, "Intrinsic disorder in proteins associated with neurodegenerative diseases," *Front Biosci*, vol. 14, pp. 5188–5238, 2009.

[10] P. Neudecker, P. Robustelli, A. Cavalli, P. Walsh, P. Lundstrm, A. Zarrine-Afsar, S. Sharpe, M. Vendruscolo, and L. E. Kay, "Structure of an intermediate state in protein folding and aggregation," *Science*, vol. 336, no. 6079, pp. 362–366, 2012.

[11] M. Karplus and J. Kuriyan, "Molecular dynamics and protein function," *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 19, pp. 6679–6685, 2005.

[12] C. B. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[13] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucl. Acids Res.*, vol. 25, no. 17, pp. 3389–33 402, 1997.

[14] A. Bairoch, P. Bucher, and K. Hoffmann, "The PROSITE database, its status in 1997," *Nucl. Acids Res.*, vol. 25, no. 1, pp. 217–221, 1997.

[15] N. Hulo, C. J. Sigrist, V. Le Saux, P. S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. De Castro, P. Bucher, and A. Bairoch, "Recent improvements to the PROSITE database," *Nucl. Acids Res.*, vol. 32, no. 1, pp. D134–D137, 2003.

[16] E. L. Sonnhammer, S. R. Eddy, and R. Durbin, "Pfam: a comprehensive database of protein domain families based on seed alignments," *Proteins: Struct. Funct. Bioinf.*, vol. 28, no. 3, pp. 405–420, 1997.

[17] E. L. Sonnhammer, S. R. Eddy, E. Birney, A. Bateman, and R. Durbin, "Pfam: Multiple sequence alignments and HMM-profiles of protein domains," *Nucl. Acids Res.*, vol. 26, no. 1, pp. 320–322, 1998.

[18] S. E. Brenner and M. Levitt, "Expectations from structural genomics," *Protein Sci.*, vol. 9, no. 1, pp. 197–200, 2000.

[19] D. Lee, O. Redfern, and C. Orengo, "Predicting protein function from sequence and structure," *Nat. Rev. Mol. Cell Biol.*, vol. 8, no. 12, pp. 995–1005, 2007.

[20] M. F. Perutz, M. G. Rossmann, A. F. Cullis, H. Muirhead, G. Will, and A. C. T. North, "Structure of myoglobin: a three-dimensional fourier synthesis at 5.5 angstrom resolution," *Nature*, vol. 185, p. 416422, 1960.

[21] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998.

[22] T. Jaakkola, M. Diekhans, and D. Haussler, "Using the fisher kernel method to detect remote protein homologies," in *Int Conf Intell Sys Mol Biol (ISMB)*, T. Lengauer, R. Schneider, P. Bork, D. Brutlag, J. Glasgow, H.-W. Mewes, and R. Zimmer, Eds. Menlo Park, CA: AAAI Press, 1999, pp. 149–159.

[23] L. Liao and W. S. Noble, "Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships," *J. Comp. Biol.*, vol. 10, no. 6, pp. 857–868, 2002.

[24] S. R. Eddy, "Hidden Markov models," *Curr. Opinion Struct. Biol.*, vol. 6, no. 3, pp. 361–365, 1995.

[25] H. M. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide Protein Data Bank," *Nat. Struct. Biol.*, vol. 10, no. 12, pp. 980–980, 2003.

[26] E. Schroedinger, *What is life?* Cambridge University Press, 1944.

[27] R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman Lectures on Physics.* Reading, MA: Addison-Wesley, 1963.

[28] Y. J. Huang and G. T. Montellione, "Structural biology: proteins flex to function," *Nature*, vol. 438, no. 7064, pp. 36–37, 2005.

[29] M. Vendruscolo and C. M. Dobson, "Dynamic visions of enzymatic reactions," *Science*, vol. 313, no. 5793, pp. 1586–1587, 2006.

[30] K. Jenzler-Wildman and D. Kern, "Dynamic personalities of proteins," *Nature*, vol. 450, pp. 964–972, 2007.

[31] J. S. Hub and B. L. de Groot, "Detection of functional modes in protein dynamics," *PLoS Comp Biol*, vol. 5, no. 8, p. e1000480, 2009.

[32] O. Beckstein, E. J. Denning, J. R. Perilla, and T. B. Woolf, "Zipping and unzipping of adenylate kinase: atomistic insights into the ensemble of open-closed transitions," *J. Mol. Biol.*, vol. 394, no. 1, pp. 160–176, 2009.

[33] R. A. Engh and R. Huber, "Accurate bond and angle parameters for X-ray protein structure refinement," *Acta Crystallogr.*, vol. A47, pp. 392–400, 1991.

[34] J. N. Onuchic and P. G. Wolynes, "Theory of protein folding," *Curr. Opinion Struct. Biol.*, vol. 14, pp. 70–75, 1997.

[35] J. N. Onuchic, Z. Luthey-Schulten, and P. G. Wolynes, "Theory of protein folding: the energy landscape perspective," *Annual Review of Physical Chemistry*, vol. 48, pp. 545–600, 1997.

[36] K. A. Dill and H. S. Chan, "From levinthal to pathways to funnels," *Nat. Struct. Biol.*, vol. 4, no. 1, pp. 10–19, 1997.

[37] J. Moult, K. Fidelis, A. Kryshtafovych, and A. Tramontano, "Critical assessment of methods of protein structure prediction (CASP) round IX," *Proteins: Struct. Funct. Bioinf.*, vol. Suppl, no. 10, pp. 1–5, 2011.

[38] A. Verma, A. Schug, K. H. Lee, and W. Wenzel, "Basin hopping simulations for all-atom protein folding," *J. Chem. Phys.*, vol. 124, no. 4, p. 044515, 2006.

[39] G. R. Bowman and V. S. Pande, "Simulated tempering yields insight into the low-resolution rosetta scoring functions," *Proteins: Struct. Funct. Bioinf.*, vol. 74, no. 3, pp. 777–788, 2009.

[40] R. Das, "Four small puzzles that rosetta doesn't solve." *PLoS ONE*, vol. 6, no. 5, p. e20044, 2011.

[41] A. Shmygelska and M. Levitt, "Generalized ensemble methods for de novo structure prediction," *Proc. Natl. Acad. Sci. USA*, vol. 106, no. 5, pp. 94 305–95 126, 2009.

[42] L. Sutto, J. Latzer, J. A. Hegler, D. U. Ferreiro, and P. G. Wolynes, "Consequences of localized frustration for the folding mechanism of the im7 protein," *Proc. Natl. Acad. Sci. USA*, vol. 104, no. 50, pp. 19 825–19 830, 2007.

[43] K. W. Kaufmann, G. H. Lemmon, S. L. DeLuca, J. H. Sheehan, and J. Meiler, "Practically useful: What the rosetta protein modeling suite can do for you," *Biochemistry*, vol. 49, no. 14, pp. 2987–2998, 2010, pMID: 20235548. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/bi902153g

[44] G. A. Papoian, J. Ulander, M. P. Eastwood, Z. Luthey-Schulten, and P. G. Wolynes, "Water in protein structure prediction," *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 10, pp. 3352–3357, 2004.

[45] A. Shehu and B. Olson, "Guiding the search for native-like protein conformations with an ab-initio tree-based exploration," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1106–11 227, 2010.

[46] B. Olson, K. Molloy, and A. Shehu, "In search of the protein native state with a probabilistic sampling approach," *J. Bioinf. and Comp. Biol.*, vol. 9, no. 3, pp. 383–398, 2011.

[47] B. S. Olson, K. Molloy, S.-F. Hendi, and A. Shehu, "Guiding search in the protein conformational space with structural profiles," *J. Bioinf. and Comput. Biol.*, vol. 10, no. 3, p. 1242005, 2012.

[48] A. Shehu, L. E. Kavraki, and C. Clementi, "Multiscale characterization of protein conformational ensembles," *Proteins: Struct. Funct. Bioinf.*, vol. 76, no. 4, pp. 837–851, 2009.

[49] J. A. Hegler, J. Laetzer, A. Shehu, C. Clementi, and P. G. Wolynes, "Restriction vs. guidance: fragment assembly and associative memory hamiltonians for protein structure prediction," *Proc. Natl. Acad. Sci. USA*, vol. 106, no. 36, pp. 15 302–15 307, 2009.

[50] D. A. Case, T. A. Darden, T. E. I. Cheatham, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, K. M. Merz, D. A. Pearlman, M. Crowley, R. C. Walker, W. Zhang, B. Wang, S. Hayik, A. Roitberg, G. Seabra, K. F. Wong, F. Paesani, X. Wu, S. Brozell, V. Tsui, H. Gohlke, L. Yang, C. Tan, J. Mongan, V. Hornak, G. Cui, P. Beroza, D. H. Mathews, C. Schafmeister, W. S. Ross, and P. A. Kollman, "Amber 9," University of California, San Francisco, 2006.

[51] H. Gong, P. J. Fleming, and G. D. Rose, "Building native protein conformations from highly approximate backbone torsion angles," *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 45, pp. 16 227–16 232, 2005.

[52] M. C. Prentiss, C. Hardin, M. P. Eastwood, C. Zong, and P. G. Wolynes, "Protein structure prediction: the next generation," *J. Chem. Theory Comput.*, vol. 2, no. 3, pp. 705–716, 2006.

[53] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker, "Protein structure prediction using rosetta," *Methods Enzymol.*, vol. 383, pp. 66–93, 2004.

[54] M. Zhang and L. E. Kavraki, "A new method for fast and accurate derivation of molecular conformations," *Chem. Inf. Comput. Sci.*, vol. 42, no. 1, pp. 64–70, 2002.

[55] A. D. McLachlan, "A mathematical procedure for superimposing atomic coordinates of proteins," *Acta Crystallogr. A.*, vol. 26, no. 6, pp. 656–657, 1972.

[56] V. N. Maiorov and G. M. Crippen, "Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins," *J. Mol. Biol.*, vol. 235, no. 2, pp. 625–634, 1994.

[57] N. Haspel, M. Moll, M. L. Baker, W. Chiu, and K. L. E., "Tracing conformational changes in proteins," *BMC Struct. Biol.*, vol. 10, no. Suppl1, p. S1, 2010.

[58] K. Molloy, J. M. Van, D. Barbará, and A. Shehu, "Higher-order representations for automated organization of protein structure space," in *IEEE International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*, New Orleans, LA, June 2013.

[59] K. Molloy, J. V. Min, D. Barbara, and A. Shehu, "Exploring representations of protein structure for automated remote homology detection and mapping of protein structure space," *BMC Bioinf*, vol. 15, no. Suppl 8, p. S4, 2014.

[60] A. E. Todd, R. L. Marsden, J. M. Thornton, and C. A. Orengo, "Progress of structural genomics initiatives: an analysis of solved target structures," *J. Mol. Biol.*, vol. 348, pp. 1235–1260, 2005.

[61] A. Godzik, "The structural alignment between two proteins: is there a unique answer?" *Protein Sci.*, vol. 5, no. 7, pp. 1325–1338, 1996.

[62] A. Stark, S. Sunyaev, and R. B. Russell, "A model for statistical significance of local similarities in structure," *J. Mol. Biol.*, vol. 326, no. 5, pp. 1307–1316, 2003.

[63] M. L. Sierk and W. R. Pearson, "Sensitivity and selectivity in protein structure comparison," *Protein Sci.*, vol. 13, no. 3, pp. 773–785, 2004.

[64] W. R. Tayor and C. A. Orengo, "Protein structure alignment," *J. Mol. Biol.*, vol. 208, pp. 1–22, 1989.

[65] W. R. Taylor and C. A. Orengo, "A holistic approach to protein structure alignment," *Protein Eng.*, vol. 2, no. 7, pp. 505–519, 1989.

[66] W. R. Taylor, "Protein structure comparison using iterated dynamic programming," *Protein Sci.*, vol. 8, no. 3, pp. 654–665, 1999.

[67] C. A. Orengo and W. R. Taylor, "SSAP: sequential structure alignment program for protein structure comparison," *Methods Enzymol*, vol. 266, pp. 617–635, 1996.

[68] G. J. Kleywegt, "Use of noncrystallographic symmetry in protein structure refinement," *Acta Crystallogr D.*, vol. 52, no. Pt. 4, pp. 842–857, 1996.

[69] M. Levitt and M. Gerstein, "A unified statistical framework for sequence comparison and structure comparison," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 11, pp. 5913–5920, 1998.

[70] S. Subbiah, D. V. Laurents, and M. Levitt, "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions," *Curr Biol*, vol. 3, no. 3, pp. 141–148, 1993.

[71] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices," *jmb*, vol. 233, no. 1, pp. 123–138, 1993.

[72] I. N. Shindyalov and P. E. Bourne, "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path," *Protein Eng.*, vol. 11, no. 9, pp. 739–747, 1998.

[73] A. Zemla, "LGA: a method for finding 3D similarities in protein structures," *Nucl. Acids Res.*, vol. 31, no. 13, pp. 3370–3374, 2003, http://as2ts.llnl.gov/AS2TS/LGA/lga.html.

[74] Y. Zhang and J. Skolnick, "TM-align: a protein structure alignment algorithm based on the TM-score," *Nucl. Acids Res.*, vol. 33, no. 7, pp. 2302–2309, 2005.

[75] T. Madej, J. F. Gibrat, and S. H. Bryant, "Threading a database of protein cores," *Proteins: Struct. Funct. Bioinf.*, vol. 23, no. 3, pp. 356–369, 1995.

[76] J. F. Gibrat, T. Madej, and S. H. Bryant, "Suprising similarities in structure comparison," *Curr. Opinion Struct. Biol.*, vol. 6, no. 3, pp. 377–385, 1996.

[77] E. Kissinel and K. Henrick, "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions," *Acta Crystallographica D Bio Crystallogr*, vol. 60, no. 12.1, pp. 2256–2268, 2004.

[78] I. Budowski-Tal, , Y. Nov, and R. Kolodny, "Fragbag, an accurate representation of protein structure, retrieves structural neighbors from the entire PDB quickly and accurately," *Proc. Natl. Acad. Sci. USA*, vol. 107, pp. 3481–3486, 2010.

[79] J. Hou, J. S.-R., C. Zhang, and S. Kim, "Global mapping of the protein structure space and application in structure-based inference of protein function," *Proc. Natl. Acad. Sci. USA*, vol. 102, pp. 3651–3656, 2005.

[80] O. Carugo, "Rapid methds for comparing protein structures and scanning structure databases," *Current Bioinformatics*, vol. 1, pp. 75–83, 2006.

[81] A. C. Martin, "The ups and downs of protein topology; rapid comparison of protein structure," *Protein Eng.*, vol. 13, no. 12, pp. 829–837, 2000.

[82] S. Kirilova and O. Carugo, "Progress in the PRIDE technique for rapidly comparing protein three-dimensional structures," *BMC Research Notes*, vol. 1, p. 44, 2008.

[83] Z. Aung and K. L. Tan, "Rapid 3D protein structure database searching using information retrieval techniques," *Bioinformatics*, vol. 20, no. 7, pp. 1045–1052, 2004.

[84] M. Carpentier, S. Brouillet, and J. Pothier, "YAKUSA: a fast structural database scanning method," *Proteins: Struct. Funct. Bioinf.*, vol. 61, no. 1, pp. 137–151, 2005.

[85] A. M. Lisewski and O. Lichtarge, "Rapid detection of similarity in protein structure and function through contact metric distances," *Nucl. Acids Res.*, vol. 34, no. 22, p. e152, 2006.

[86] Z. H. Zhang, K. L. Hwee, and I. Mihalek, "Reduced representation of protein structure: implications on efficiency and scope of detection of structural similarity," *BMC Bioinformatics*, vol. 11, p. 155, 2010.

[87] P. Rogen and B. Fain, "Automatic classification of protein structure by using gauss integrals," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 1, pp. 119–124, 2003.

[88] O. Carugo and S. Pongor, "Protein fold similarity estimated by a probabilistic approach based on c(α)-c(α) distance comparison," *J. Mol. Biol.*, vol. 315, no. 4, pp. 887–898, 2002.

[89] R. Kolodny, P. Koehl, L. Guibas, and M. Levitt, "Small libraries of protein fragments model native protein structures accurately," *J. Mol. Biol.*, vol. 323, no. 2, pp. 297–307, 2002.

[90] S. M. Salem, M. J. Zaki, and C. Bystroff, "Flexible non-sequential protein structure alignment," *Algorithms for Molecular Biology*, vol. 5, no. 1, p. 12, 2010.

[91] Y. Ye and A. Godzik, "Flexible stucture alignment by chaining aligned fragment pairs allowing twists," *Bioinformatics*, vol. 19, no. 2, pp. 246–255, 2003.

[92] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York: Cambridge University Press, 2008.

[93] M. Osadchy and R. Kolodny, "Maps of protein structure space reveal a fundamental relationship between protein structure and function," *Proc. Natl. Acad. Sci. USA*, vol. 108, pp. 12 301–12 306, 2011.

[94] D. M. Blei, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[95] S. Kullback, "Letter to the editor: The kullback-leibler distance," *The American Statistician*, vol. 41, pp. 340–341, 1987.

[96] G. W. Corder and D. I. Foreman, *Nonparametric statistics for non-statisticians: A step-by-step approach*. New York: Wiley, 2009.

[97] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag, 1995.

[98] M. Steyvers and T. Griffiths, "Probabilistic topic models," in *Latent Semantic Analysis: A Road to Meaning.*, T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, Eds. Hillsdate, NJ: Laurence Erlbaum, 2006. [Online]. Available: http://cocosci.berkeley.edu/tom/papers/SteyversGriffiths.pdf

[99] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[100] Schrödinger, LLC, "The PyMOL molecular graphics system, version 1.3r1," August 2010.

[101] K. Molloy and A. Shehu, "Biased decoy sampling to aid the selection of near-native protein conformations," in *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, ser. BCB '12. New York, NY, USA: ACM, 2012, pp. 131–138. [Online]. Available: http://doi.acm.org/10.1145/2382936.2382953

[102] K. Molloy, S. Saleh, and A. Shehu, "Probabilistic search and energy guidance for biased decoy sampling in ab-initio protein structure prediction," *IEEE/ACM Trans Bioinf and Comp Biol*, 2013, in press.

[103] J. Lee, S. Wu, and Y. Zhang, "Ab initio protein structure prediction," in *Ab Initio Protein Structure Prediction*, D. Rigden, Ed. Springer Science + Business Media B.V., 2009, ch. 1.

[104] Y. Zhang, "Progress and challenges in protein structure prediction," *Curr. Opinion Struct. Biol.*, vol. 18, no. 3, pp. 342–348, 2008.

[105] A. Shehu, "Conformational search for the protein native state," in *Protein Structure Prediction: Method and Algorithms*, H. Rangwala and G. Karypis, Eds. Fairfax, VA: Wiley Book Series on Bioinformatics, 2010, ch. 21.

[106] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[107] K. F. Han and D. Baker, "Global properties of the mapping between local amino acid sequence and local structure in proteins," *Proc. Natl. Acad. Sci. USA*, vol. 93, no. 12, pp. 5814–5818, 1996.

[108] B. Olson, K. Molloy, and A. Shehu, "Enhancing sampling of the conformational space near the protein native state," in *LNCS-BIONETICS: Intl. Conf. on Bio-inspired Models of Network, Information, and Computing Systems*, vol. 87, Boston, MA, December 2010, pp. 249–263.

[109] J. Handl, J. Knowles, R. Vernon, D. Baker, and S. C. Lovell, "The dual role of fragments in fragment-assembly methods for de novo protein structure prediction," *Proteins: Struct. Funct. Bioinf.*, vol. 80, no. 2, pp. 490–504, 2011.

[110] P. Bradley, K. M. S. Misura, and D. Baker, "Toward high-resolution de novo structure prediction for small proteins," *Science*, vol. 309, no. 5742, pp. 1868–1871, 2005.

[111] D. Xu and Y. Zhang, "Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field," *Proteins: Struct. Funct. Bioinf.*, vol. 80, no. 7, pp. 1715–1735, 2012.

[112] M. Stilman and J. J. Kuffner, "Planning among movable obstacles with artificial constraints," *Int. J. Robot. Res.*, vol. 12, no. 12, pp. 1295–1307, 2008.

[113] E. Plaku, L. Kavraki, and M. Vardi, "Discrete search leading continuous exploration for kinodynamic motion planning," in *Robotics: Sci. and Syst.*, Atlanta, GA, USA, 2007.

[114] Y. Yang and O. Brock, "Efficient motion planning based on disassembly," in *Robotics: Sci. and Syst.*, Cambridge, MA, 2005, pp. 97–104.

[115] H. Kurniawati and D. Hsu, "Workspace-based connectivity oracle: An adaptive sampling strategy for PRM planning," in *WAFR*, ser. Springer Tracts in Advanced Robotics, New York, NY, 2006, vol. 47, pp. 35–51.

[116] J. P. van den Berg and M. H. Overmars, "Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners," *Int. J. Robot. Res.*, vol. 24, no. 12, pp. 1055–1071, 2005.

[117] P. J. Ballester, I. Westwood, N. Laurieri, E. Sim, and W. G. Richards, "Prospective virtual screening with ultrafast shape recognition: the identification of novel inhibitors of arylamine n-acetyltransferases," *Journal of The Royal Society Interface*, vol. 7, no. 43, pp. 335–342, 2010. [Online]. Available: http://rsif.royalsocietypublishing.org/content/7/43/335.abstract

[118] B. Olson, S. Hendi, K. Molloy, and A. Shehu, "Protein conformational search with geometric projections," in *IEEE BIBMW - Comput Struct Biol Workshop (CSBW)*, Atlanta, GA, November 2011, pp. 366–373.

[119] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *In KDD Workshop on Text Mining*, 2000.

[120] Y. Zhang and J. Skolnick, "Spicker: a clustering approach to identify near-native protein folds," *J. Comput. Chem.*, vol. 25, p. 865871, 2004.

[121] M. C. Prentiss, D. J. Wales, and P. G. Wolynes, "Protein structure prediction using basin-hopping." *The Journal of Chemical Physics*, vol. 128, no. 22, pp. 225 106–225 106, Jun. 2008.

[122] K. Molloy and A. Shehu, "A robotics-inspired method to sample conformational paths connecting known functionally-relevant structures in protein systems," in *IEEE BIBMW - Comput Struct Biol Workshop (CSBW)*, J. He, A. Shehu, N. Haspel, and C. B., Eds., Philadelphia, PA, October 2012, pp. 56–63.

[123] P. Majek, H. Weinstein, and R. Elber, *Pathways of conformational transitions in proteins*. Taylor and Francis group, 2008, ch. 13, pp. 185–203.

[124] T. Hansson, C. Oostenbrink, and W. F. van Gunsteren, "Molecular dynamics simulations," *Curr. Opinion Struct. Biol.*, vol. 12, no. 2, pp. 190–196, 2002.

[125] M. Karplus and J. A. McCammon, "Molecular dynamics simulations of biomolecules," *Nat. Struct. Biol.*, vol. 9, no. 9, pp. 646–652, 2002.

[126] H. Huang, E. Ozkirimli, and C. B. Post, "A comparison of three perturbation molecular dynamics methods for mmodeling conformational transitions," *J. Chem. Theory Comput.*, vol. 5, no. 5, pp. 1301–1314, 2009.

[127] R. Malek and N. Mousseau, "Dynamics of Lennard-Jones clusters: a characterization of the activation-relaxation technique," *Phys. Rev. E*, vol. 62, no. 6, pp. 7723–7728, 2000.

[128] D. J. Earl and M. W. Deem, "Parallel tempering: theory, applications, and new perspectives," *Phys. Chem. Chem. Phys.*, vol. 7, pp. 3910–3916, 2005.

[129] K. Arora and C. L. I. Brooks, "Large-scale allosteric conformational transitions of adenylate kinase appear to involve a population-shift mechanism," *Proc. Natl. Acad. Sci. USA*, vol. 104, no. 47, pp. 18 496–18 501, 2007.

[130] Y. Zhang, D. Kihara, and J. Skolnick, "Local energy landscape flattening: parallel hyperbolic Monte Carlo sampling of protein folding," *Proteins: Struct. Funct. Bioinf.*, vol. 48, no. 2, pp. 192–201, 2002.

[131] 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," *J. Am. Chem. Soc.*, vol. 122, no. 36, pp. 8700–8711, 2000.

[132] P. Krueger, S. Verheyden, P. J. Declerck, and Y. Engelborghs, "Extending the capabilities of targeted molecular dynamics: simulation of a large conformational transition in plasminogen activator inhibitor 1," *Protein Sci.*, vol. 10, no. 4, pp. 798–808, 2001.

[133] J. Schlitter, M. Engels, and P. Krueger, "Targeted molecular dynamics - a new approach for searching pathways of conformational transitions," *Proteins: Struct. Funct. Bioinf.*, vol. 12, no. 2, pp. 84–89, 1994.

[134] R. J. Mashi and E. Jakobsson, "End-point targeted molecular dynamics: large-scale conformational changes in potassium channels," *Biophys. J.*, vol. 94, no. 11, pp. 4307–4319, 2008.

[135] A. van der Vaart and M. Karplus, "Minimum free energy pathways and free energy profiles for conformational transitions based on atomistic molecular dynamics simulations," *J. Chem. Phys.*, vol. 126, p. 164106, 2007.

[136] A. C. Pan, D. Sezer, and B. Roux, "Finding transition pathways using the string method with swarms of trajectories," *J. Phys. Chem. B*, vol. 112, no. 11, pp. 3432–3440, 2008.

[137] B. W. Zhang, D. Jasnow, and D. M. Zuckermann, "Efficient and verified simulation of a path ensemble for conformational change in a united-residue model of calmodulin," *Proc. Natl. Acad. Sci. USA*, vol. 104, no. 46, pp. 18 043–18 048, 2007.

[138] K. M. Kim, R. L. Jernigan, and G. S. Chirikjian, "Efficient generation of feasible pathways for protein conformationa transitions," *Biophys. J.*, vol. 83, no. 3, pp. 1620–1630, 2002.

[139] A. d. Schuyler, R. L. Jernigan, P. K. Wasba, B. Ramakrishnan, and G. S. Chirikjian, "Iterative cluster-nma (icnma): a tool for generating conformational transitions in proteins," *Proteins: Struct. Funct. Bioinf.*, vol. 74, no. 3, pp. 760–776, 2009.

[140] W. Zheng and B. Brooks, "Identification of dynamical correlations within the myosin motor domain by the normal mode analysis of an elastic network model," *J. Mol. Biol.*, vol. 346, no. 3, pp. 745–759, 2005.

[141] R. Bahar and A. J. Rader, "Coarse-grained normal mode analysis in structural biology," *Curr. Opinion Struct. Biol.*, vol. 204, no. 5, pp. 1–7, 2005.

[142] N. Kantarci-Carsibasi, T. Haliloglu, and P. Doruker, "Conformational transition pathways explored by monte carlo simulation integrated with collective modes," *Biophys. J.*, vol. 95, no. 12, pp. 5862–5873, 2008.

[143] A. Korkut and W. A. Hendrickson, "Computation of conformational transtions in proteins by virtual atom molecular mechanics as validated in application to adenylate kinase," *Proc. Natl. Acad. Sci. USA*, vol. 106, no. 37, pp. 15 673–15 678, 2009.

[144] M. Teknipar and W. Zheng, "Predicting order of conformational changes during protein conformational transitions using an interpolated elastic network model," *Proteins: Struct. Funct. Bioinf.*, vol. 78, no. 11, pp. 2469–2481, 2010.

[145] S. Kirillova, J. Cortés, A. Stefaniu, and T. Simeon, "An nma-guided path planning approach for computing large-amplitude conformational changes in proteins," *Proteins: Struct. Funct. Bioinf.*, vol. 70, no. 1, pp. 131–143, 2008.

[146] H. Lou and R. I. Wang, "Molecular dynamics of apo-adenylate kinase: a distance replica exchange method for the free energy of conformational fluctuations," *J. Phys. Chem. B*, vol. 110, no. 47, pp. 24 121–24 137, 2006.

[147] M. B. Kuniztki and B. L. de Groot, "The atomistic mechanism of conformational transition in adenylate kinase: a TEE-REX molecular dynamics study," *Structure*, vol. 16, no. 8, pp. 1175–1182, 2008.

[148] J. W. Chu, B. L. Trout, and C. L. I. Brooks, "A super-linear minimization scheme for the nudged elastic band method," *J. Chem. Phys.*, vol. 119, pp. 12 708–12 717, 2003.

[149] L. Maragliano, A. Fiser, E. J. Vanden-Eijnden, and G. Ciccotti, "String method in collective variables: minimum free energy paths and isocommittor surfaces," *J. Chem. Phys.*, vol. 125, p. 024106, 2006.

[150] E. Weinan, W. Ren, and E. Vanden-Eijnden, "Simplified and improved string method for computing the minimum energy paths in barrier-crossing events," *J. Chem. Phys.*, vol. 126, p. 164103, 2007.

[151] L. Maragliano and E. Vanden-Eijnden, "On-the-fly string method for minimum free energy paths calculation," *Chem. Phys. Lett.*, vol. 446, pp. 182–190, 2007.

[152] E. Weinan, W. Ren, and E. Vanden-Eijnden, "Finite temperature string methods for the study of rare events," *J. Phys. Chem.*, vol. 109, pp. 6688–6693, 2005.

[153] W. Ren, E. Vanden-Eijnden, P. Maragakis, and E. Weinan, "Transition pathways in complex systems: application of the finite-temperature string method to the alanine dipeptide," *J. Chem. Phys.*, vol. 123, p. 134109, 2005.

[154] D. R. Weiss and M. Levitt, "Can morphing methods predict intermediate structures?" *J. Mol. Biol.*, vol. 385, no. 2, pp. 665–674, 2009.

[155] D. D. Boehr and P. E. Wright, "How do proteins interact?" *Science*, vol. 320, no. 5882, pp. 1429–1430, 2008.

[156] D. D. Boehr, R. Nussinov, and P. E. Wright, "The role of dynamic conformational ensembles in biomolecular recognition," *Nature Chem Biol*, vol. 5, no. 11, pp. 789–96, 2009.

[157] L. E. Kavraki, P. Svetska, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[158] H. Choset and et al., *Principles of Robot Motion: Theory, Algorithms, and Implementations*, 1st ed. Cambridge, MA: MIT Press, 2005.

[159] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[160] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 233–255, 2002.

[161] A. M. Ladd and L. E. Kavraki, "Motion planning in the presence of drift, underactuation and discrete system changes," in *Robotics: Sci. and Syst.*, Boston, MA, 2005, pp. 233–241.

[162] A. Shehu, C. Clementi, and L. E. Kavraki, "Modeling protein conformational ensembles: From missing loops to equilibrium fluctuations," *Proteins: Struct. Funct. Bioinf.*, vol. 65, no. 1, pp. 164–179, 2006.

[163] J. Cortés, T. Simeon, M. Remauld-Simeon, and V. Tran, "Geometric algorithms for the conformational analysis of long protein loops," *J. Comput. Chem.*, vol. 25, no. 7, pp. 956–967, 2004.

[164] J. Cortés, D. Le, R. Lehl, and T. Simeon, "Simulating ligand-induced conformational changes in proteins using a mechanical disassembly method," *Phsy. Chem. Chem. Phys.*, vol. 12, no. 29, pp. 8268–8276, 2010.

[165] D. Devaurs, L. Bouard, M. Vaisset, C. Zanon, I. Al-Bluwi, I. R., T. Sim'eon, and J. Cort'es, "Moma-ligpath: a web server to simulate protein-ligand unbinding," *Nucl. Acids Res.*, vol. 41, pp. W297–W302, 2013.

[166] J. Cortés, T. Simeon, R. de Angulo, D. Guieysse, M. Remaud-Simeon, and V. Tran, "A path planning approach for computing large-amplitude motions of flexible molecules," *Bioinformatics*, vol. 21, no. S1, pp. 116–125, 2005.

[167] L. Jaillet, F. J. Corcho, J.-J. Perez, and J. Cortés, "Randomized tree construction algorithm to explore energy landscapes," *J. Comput. Chem.*, vol. 32, no. 16, pp. 3464–3474, 2011.

[168] B. Gipson, M. Moll, and L. E. Kavraki, "Sims: A hybrid method for rapid conformational analysis," *PLoS ONE*, vol. 8, no. 7, p. e68826, 07 2013.

[169] E. Project, R. Friedman, E. Nachliel, and M. Gutman, "A molecular dynamics study of the effect of Ca^{2+} removal on calmodulin structure," *Biophys. J.*, vol. 90, no. 11, pp. 3842–3850, 2006.

[170] B. E. Finn, J. Evenäs, T. Drakenberg, J. P. Waltho, E. Thulin, and S. Forsén, "Calcium-induced structural changes and domain autonomy in calmodulin," *Nat. Struct. Biol.*, vol. 2, no. 9, pp. 777–783, 1995.

[171] J. Evenäs, S. Forsén, A. Malmendal, and M. Akke, "Backbone dynamics and energetics of a calmodulin domain mutant exchanging between closed and open conformations," *J. Mol. Biol.*, vol. 289, no. 3, pp. 603–617, 1999.

[172] S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman, "The weighted histogram analysis method for free-energy calculations on biomolecules: I. The method," *J. Comput. Chem.*, vol. 13, no. 8, pp. 1011–1021, 1993.

[173] K. P. Ravindranathan, E. Gallicchio, and R. M. Levy, "Conformational equilibria and free energy profiles for the allosteric transition of the ribose-binding protein," *J. Mol. Biol.*, vol. 353, no. 1, pp. 196–210, 2005.

[174] K. Arnold, L. Bordoli, J. Kopp, and T. Schwede, "The swiss-model workspace: a web-based environment for protein structure homology modelling," *Bioinformatics*, vol. 22, no. 2, pp. 195–201, 2006.

[175] M. K. and S. A., "On the stochastic roadmap to model functionally-related structural transitions in wildtype and variant proteins," in *RSS - Workshop on Robotic Methods for Structural and Dynamic Modeling of Molecular Systems)*, Berkeley, CA, July 2014.

[176] L. E. Kavraki, P. Svetska, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transaction on Robotics and Automation*, vol. 12, pp. 566–580, 1996.

[177] A. P. Singh, J.-C. Latombe, and D. L. Brutlag, "A motion planning approach to flexible ligand binding," in *Proc Int Conf Intell Sys Mol Biol (ISMB)*, R. Schneider, P. Bork, D. L. Brutlag, J. I. Glasgow, H.-W. Mewes, and R. Zimmer, Eds., vol. 7. Heidelberg, Germany: AAAI, 1999, pp. 252–261.

[178] 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. Biol.*, vol. 10, no. 3-4, pp. 239–255, 2002.

[179] S. Thomas, G. Song, and N. Amato, "Protein folding by motion planning," *Physical Biology*, no. 2, pp. S148–S155, 2005.

[180] 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," *J. Comp. Biol.*, vol. 10, no. 3-4, pp. 257–281, 2003.

[181] R. Clausen and A. Shehu, "Multiscale hybrid evolutionary algorithm to obtain sample-based representations of multi-basin protein energy landscapes," in *ACM Conf on Bioinf and Comp Bio(BCB)*, Newport Beach, CA, September 2014.

[182] J. Hartigan, *Clustering Algorithms*. New York: John Wiley and Sons, 1975.

[183] R. Bohlin and L. E. Kavraki, "Path planning using lazy PRM," in *IEEE Intl. Conf. on Robotics and Automation.* San Francisco, CA: IEEE, 2000, pp. 521–528.

[184] Y. JY, "Finding the k shortest loop less paths in a network." *Management Science*, vol. 17, pp. 712–716, 1971.

[185] A. E. Karnoub and R. A. Weinberg, "Ras oncogenes: split personalities," *Nature Reviews Molecular Cell Biology*, vol. 9, pp. 517–531, 2008.

[186] R. Clausen and A. Shehu, "Exploring the structure space of wildtype ras guided by experimental data," in *ACM BCBMW - Comput Struct Biol Workshop (CSBW)*, Washington, D.C., September 2013, pp. 757–764.

[187] A. A. Gorfe, B. J. Grant, and J. A. McCammon, "Mapping the nucleotide and isoform-dependent structural and dynamical features of Ras proteins," *Structure*, vol. 16, no. 6, pp. 885–896, 2008.

[188] C. Clementi, "Coarse-grained models of protein folding: Toy-models or predictive tools?" *Curr. Opinion Struct. Biol.*, vol. 18, pp. 10–15, 2008.

[189] M. A. Rohrdanz, W. Zheng, and C. Clementi, "Discovering mountain passes via torchlight: methods for the definition of reaction coordinates and pathways in complex macromolecular reactions," vol. 64, pp. 295–316, 2013.

[190] R. Pearce, M. Morales, and N. Amato, "Structural improvement filtering strategy for PRM," in *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008.

[191] C. Nielsen and L. Kavraki, "A two level fuzzy prm for manipulation planning," in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3, 2000, pp. 1716–1721 vol.3.

[192] S. Fischer and M. Karplus, "Conjugate peak refinement: an algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom," *Chemical Physics Letters*, vol. 194, no. 3, pp. 252–261, Jun. 1992. [Online]. Available: http://dx.doi.org/10.1016/0009-2614(92)85543-j

[193] S. Brin, "Near neighbor search in large metric spaces," 1995, pp. 574–584.

[194] I. Al-Bluwi, M. Vaisset, T. Siméon, and J. Cortés, "Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods," *BMC Structural Biology*, vol. 13, no. Suppl 1, p. S8, 2013.

[195] D. Luo and N. Haspel, "Multi-resolution rigidity-based sampling of protein conformational paths," in *CSBW (Computational Structural Bioinformatics Workshop), in proc. of ACM-BCB (ACM International conference on Bioinformatics and Computational Biology)*, September 2013, pp. 787–793.

Curriculum Vitae

Kevin Molloy graduated from Hampton Road Academy in Newport News, VA in 1989. He attended George Mason University from 1989 to 1998, graduating with a Bachelor of Science in Computer Science while maintaining a job performing database programming. After graduation, he started a small company where he designed and built data replication software from extremely large mainframe databases to relational databases operating on UNIX platforms. Mr. Molloy attended George Mason University from 2008 to 2011 and received a Masters of Science in Computer Science. Mr. Molloy is now a PhD student in the Computer Science department at George Mason University, where he is a research assistant in Dr. Amarda Shehu's Computational Biology lab.

Education

- Masters of Science, Computer Science, George Mason University, 2011
- Bachelor of Science, Computer Science, George Mason University, 1998

Awards

- Nominee for best paper award, BiCoB conference (2014)
- ACM Recognition of Service Award as Conference Volunteer for ACM-BCB (2013)
- Dean Felllowship at George Mason University (2011-2012)
- Outstanding Academic Achievement Award, M.S. Computer Science, George Mason University (2011)
- Best Student paper award (2nd author), BIONETICS conference (2010)

Journal Articles (6)

1. Kevin Molloy, M. Jennifer Van, Daniel Barbará and Amarda Shehu. Exploring Representations of Protein Structure for Automated Remote Homology Detection and Mapping of Protein Structure Space. *BMC Bioinformatics Journal* 15 (Suppl 8):S4, 2014. [Impact Factor: 3.02]
2. Kevin Molloy, Sameh Saleh, and Amarda Shehu. Probabilistic Search and Energy Guidance for Biased Decoy Sampling in Ab-initio Protein Structure Prediction. *IEEE Transactions in Computational Biology and Bioinformatics Journal* 10(5):1162-1175, 2013. [Impact Factor: 2.25] [Citations: 5]

3. Kevin Molloy and Amarda Shehu. Elucidating the Ensemble of Functionally-relevant Transitions in Protein Systems with a Robotics-inspired Method. *BMC Structural Biology Journal* 13(Suppl 1):S8, 2013. [Impact Factor: 2.09] [Citations: 3]

4. Brian Olson, Irina Hashmi, Kevin Molloy, and Amarda Shehu. Basin Hopping as a General and Versatile Optimization Framework for the Characterizations of Biological Macromolecules. *Advances in Artificial Intelligence Journal*, 674832, 2012 (special issue on Artificial Intelligence in Biomedicine). [Acceptance Rate: 9%] [Citations: 9]

5. Brian Olson, Kevin Molloy, S. Farid Hendi, Amarda Shehu. Guiding Search in the Protein Conformational Space with Structural Profiles. *Journal of Bioinformatics and Computational Biology* 10(3):1242005, 2012. [Impact Factor: 1.06] [Citations: 17]

6. Brian Olson, Kevin Molloy, Amarda Shehu. In Search of the Protein Native State with a Probabilistic Sampling Approach. *Journal of Bioinformatics and Computational Biology* 9(3):383-398, 2011. [Impact Factor: 1.06] [Citations: 24]

Conference Proceedings (5)

1. Kevin Molloy and Amarda Shehu. A Probabilistic Roadmap-based Method to Model Conformational Switching of a Protein Among Many Functionally-relevant Structures. *6th Intl Conference on Bioinformatics and Comp Biology (BiCOB)*, Las Vegas, NV, 2014 *(finalist for best paper award).*

2. Kevin Molloy, Jennifer M. Van, Daniel Barbará, and Amarda Shehu. Higher-order Representations for Automated Organization of Protein Structure Space. *IEEE International Conf. on Computational Advances in Bio and Medical Sciences (ICCABS)*, New Orleans, LA, 2013. [Acceptance Rate: 42%] [Citations: 1]

3. Kevin Molloy and Amarda Shehu. Biased Decoy Sampling to Identify Near-Native Protein Conformations. *ACM Bioinf and Comp Bio (BCB)*, Orlando, FL. 2012, pg. 131-138. [Acceptance Rate: 21%] [Citations: 2]

4. Brian Olson, Kevin Molloy, and Amarda Shehu. Enhancing Sampling of the Conformational Space Near the Protein Native State. In *Intl. Conference Bio-inspired Models of Network, Information, and Computing Systems (BIONETICS)*, LNICST (Springer), vol. 87, pg. 249-263, Boston, MA, 2010 *(best student paper award).* [Acceptance Rate: 24%] [Citations: 12]

5. Kevin Molloy and Daniel Menascé. Method and Model to Assess the Performance of Clustered Databases: The Oracle RAC Case. In *Computer Measurement Group (CMG)* , Orlando, FL. December 2010.

Workshop Articles (2)

1. Kevin Molloy, Rudy Clausen, Amarda Shehu. On the Stochastic Roadmap to Model Functionally-related Structural Transitions in Wildtype and Variant Proteins. *RSS Workshop on Robotic Methods for Structural and Dynamic Modeling of Molecular Systems*, Berkeley, CA, 2014.

2. Kevin Molloy and Amarda Shehu. A Robotics-inspired Method to Sample Conformational Paths Connecting Known Functionally-relevant Structures in Protein Systems. Comput Struct Biol Workshop (CSBW) - IEEE BIBM Workshops, pg. 56-63, Philadelphia, PA, 2012. [Acceptance Rate: 33%] [Citations: 4]