Robotics-inspired methods for the simulation of conformational changes in proteins.

Ibrahim Al Bluwi

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National des Sciences Appliquées de Toulouse (INSA Toulouse)

Discipline ou spécialité :

Systèmes Automatiques et Informatiques

Présentée et soutenue par :
Ibrahim AL BLUWI

le : mardi 25 septembre 2012

Titre :

Méthodes Inspirées de la Robotique pour la Simulation des Changements Conformationnels des Protéines

---

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)

Directeur(s) de Thèse :

Juan CORTÉS (LAAS-CNRS, Toulouse)
Thierry SIMÉON (LAAS-CNRS, Toulouse)

Rapporteurs :

Victor GUALLAR (BSC, Barcelone)
Oliver BROCK (TU, Berlin)

Membre(s) du jury :

Yves-Henri SANEJOUAND (FRE-CNRS, Nantes) - Examinateur
Alain ESTÈVE (LAAS-CNRS, Toulouse) - Examinateur

PhD Thesis

Presented at

Laboratoire d'Analyse et d'Architecture des Systémes (LAAS-CNRS)

Submitted for obtaining the degree of

Doctor of the University of Toulouse
Delivered By INSA-Toulouse

In : **Automatic Systems and Computer Science**

Ecole doctorale Systémes (EDSYS)

By : **Ibrahim Albluwi**

Robotics-Inspired Methods for the Simulation of Conformational Changes in Proteins

---

Defended on the **25^{th} of September 2012**

Committee Members:

Victor Guallar	BSC, Barcelone	***Reviewers***
Oliver Brock	TU, Berlin	
Yves-Henri Sanejouand	FRE-CNRS, Nantes	***Examiners***
Alain Estéve	LAAS-CNRS, Toulouse	
Juan Cortés	LAAS-CNRS, Toulouse	***Thesis Directors***
Thierry Siméon	LAAS-CNRS, Toulouse	

Abstract

Proteins are biological macromolecules that play essential roles in living organisms. Understanding the relationship between protein structure, dynamics and function is indispensable for advances in fields such as biology, pharmacology and biotechnology. Studying this relationship requires a combination of experimental and computational methods, whose development is the object of very active interdisciplinary research. In such a context, this thesis presents a robotics-inspired modeling approach for studying conformational changes in proteins. This approach is based on a mechanistic representation of proteins that enables the application of efficient methods originating from the field of robotics. It also provides an accurate method for coarse-grained treatment of proteins without loosing full-atom details.

The presented approach is applied in this thesis to two different molecular simulation problems. First, the approach is used to enhance sampling of the conformational space of proteins using the Monte Carlo method. The modeling approach is used to implement new and known Monte Carlo trial move classes as well as a mixed sampling strategy. Results of simulations performed on proteins with different topologies show that this strategy enhances sampling without demanding higher computational resources. In the second problem tackled in this thesis, the mechanistic modeling approach is used to implement a robotics-inspired method for simulating large amplitude motions in proteins. This method is based on the combination of the Rapidly-exploring Random Tree (RRT) algorithm with Normal Mode Analysis (NMA), which allows efficient exploration of the high dimensional conformational spaces of proteins. Results of simulations performed on ten different proteins of different sizes and topologies show the effectiveness of the proposed method for studying conformational transitions.

Résumé

Les protéines sont des macromolécules biologiques qui jouent des rôles essentiels dans les organismes vivants. La compréhension de la relation entre la structure des protéines, leur dynamique et leur fonction est indispensable pour progresser dans des domaines tels que la biologie, la pharmacologie et les biotechnologies. L'étude de cette relation exige une combinaison de méthodes expérimentales et de méthodes de calcul, dont le développement est l'objet d'une recherche interdisciplinaire très active. Dans ce contexte, cette thèse présente une approche de modélisation inspirée par la robotique pour l'étude des changements conformationnels des protéines. Cette approche est basée sur une représentation mécanistique des protéines permettant l'application de méthodes efficaces provenant du domaine de la robotique. Elle fournit également une méthode appropriée pour le traitement Ç gros-grains È des protéines sans perte de détail au niveau atomique.

L'approche présentée dans cette thèse est appliquée à deux types de problèmes de simulation moléculaire. Dans le premier, cette approche est utilisée pour améliorer l'échantillonnage de l'espace conformationnel des protéines. Plus précisément, cette approche de modélisation est utilisée pour implémenter des classes de mouvements pour l'échantillonnage, aussi bien connues que nouvelles, ainsi qu'une stratégie d'échantillonnage mixte, dans le contexte de la méthode de Monte Carlo. Les résultats des simulations effectuées sur des protéines ayant des topologies différentes montrent que cette stratégie améliore l'échantillonnage, sans toutefois nécessiter de ressources de calcul supplémentaires. Dans le deuxième type de problèmes abordés ici, l'approche de modélisation mécanistique est utilisée pour implémenter une méthode inspirée par la robotique et appliquée à la simulation de mouvements de grande amplitude dans les protéines. Cette méthode est basée sur la combinaison de l'algorithme RRT (Rapidly-exploring Random Tree) avec l'analyse en modes normaux (Normal Mode Analysis, ou NMA), qui permet une exploration efficace des espaces de dimension élevée tels les espaces conformationnels des protéines. Les résultats de simulations effectuées sur un ensemble de protéines de tailles et de topologies différentes montrent l'efficacité de la méthode proposée pour l'étude des transitions conformationnelles.

To my wonderful wife, Ameera, I dedicate this work.

Acknowledgements

The work presented in this thesis was carried out in the Robotics group at LAAS-CNRS. Many thanks to the lab directors, Raja Chatila and Jean Arlat, to the head of the group Rachid Alami, to my thesis director Thierry Siméon, and to all the group members and administrative staff of the lab.

My sincere gratitude and appreciation go to my direct supervisor Juan Cortés for his guidance, help and patience throughout my three years of study. His close and attentive supervision as well as his support that extended beyond the scientific work were a great aid during my PhD journey. Many thanks also to the thesis jury: Oliver Brock, Victor Guallar, Alain Estéve and Yves-Henri Sanejouand for their valuable comments and suggestions.

Special thanks to my colleagues Didier Devaurs, Marc Vaisset and Romain Iehl for their collaboration and generous help, which enriched my thesis and alleviated much of its stressful work. Many thanks also to my friends Mohammad Ali, Alhayat Ali, Dawood Khan, Amit Pandey, Naveed Muhammad and Lavindra De Silva for their help and encouragement and for making my stay at the lab more pleasant and enjoyable. I am deeply thankful to everyone that contributed to this thesis or helped me during my stay in France in any way.

Hearty thanks to my friends outside work: Nadim Nasreddine, Houssam Arbess, Alaa Allouch, Hadi El Bayda, Mahmoud Maksoud, Ahmad Al Sheikh, Réda Hassaine, Anouar Rachdi, Ahmad Akl, Ali Shoker, Houssem Jerbi, Alaa Fouda, Ziad Almaksour, Bashar Kabbani, Bhjat Elrez and many others. I am truly indebted to them for their unlimited help and encouragement, and for adding a beautiful taste to my stay in Toulouse.

I also would like to say that I am infinitely and wholeheartedly grateful to my parents. They have implanted in me the love of knowledge and have laid the foundations I am now standing on. They have always showered me with lavish love and care and have always tried their best to push me towards perfection. Their favors are certainly beyond repay. I am also thankful to my beloved brothers and sisters Qutaiba, Najla, Ghada and Tariq for their love and care and for their continuous encouragement and prayers.

Finally, a word of deep love and gratitude to my wife, Ameera. She was beside me during all the happy and hard moments of the PhD and sacrificed her time and energy for the sake of mine. *Thank you from all my heart.*

Contents

List of Figures

List of Tables

Introduction

Computer simulations are widely used nowadays to model biomolecules, mimic their behavior and gain insight about their physicochemical properties and biological functions. Indeed, a whole field dedicated to such simulations currently exists under the name of computational structural biology.

Computational methods have been mostly developed for complementing experimental methods. For instance, molecular dynamics (MD) [Rapaport 07] and Monte Carlo (MC) methods [Landau 05] are largely used to study thermodynamic properties and the activity of proteins from an initial structure determined by X-ray crystallography [Woolfson 97] or nuclear magnetic resonance (NMR) [Cavanagh 06]. The complementarity between experimental and computational methods can also be exploited in the other direction, since simulations can be enhanced using experimental data. An interesting illustration of that is the use of NMR chemical shifts to restrain MD simulations [Robustelli 10].

Some computational methods go further, aiming to replace experimental methods. For instance, computational methods can be used to determine the structure of proteins without prior experimental information [Bonneau 01]. Methods are also available for predicting molecular interactions (molecular docking) [Lengauer 96], and for understanding how proteins move from random coils to their native structure (protein folding) [Pain 00]. Nevertheless, the current status of these computational methods is still far from providing completely accurate and reliable results in all the cases, and the most complex instances of the aforementioned problems remain out of reach for state-of-the-art methods. For example, current computational power permits performing MD simulations that cover up to some microseconds of the physical time. This is of course insufficient since molecular motions in some events like protein folding can occur over the range of seconds [Muñoz 08]. MC methods also suffer from shortcomings in their search and sampling of the conformational space of proteins, which is a rugged landscape with many local minima. MC methods tend to get trapped in these local minima and waste considerable time trying to escape out of them.

For these reasons, active research is currently focused on enhancing simulation techniques (see [Sugita 99, Marinari 92, Laio 02, Shaw 10] for example) and producing alternatives for them. This thesis falls under a particular family of such alternative methods, which are inspired from the field of robot motion planning. Robotics-inspired methods have been introduced recently for simulating motions of proteins and for studying problems like protein folding and protein-ligand interactions. They borrow ideas, mainly, from sampling-based motion planning algorithms [LaValle 06, Choset 05, Tsianos 07], which have proven to be powerful tools for tackling high-dimensional robot motion planning problems.

Although the two fields of robotics and molecular simulations seem very distant at first glance, a closer look reveals many similarities in terms of the formulation of the tackled problems. In an early survey [Parsons 94], Parsons and Canny have shown that several of the problems studied in the field of computational structural biology are actually geometric problems that have counterparts in the field of robotics. This is mainly due to the fact that motion plays a central role for both robots and proteins. Indeed, molecular motions make an integral part of the biological processes proteins are involved in, such as catalysis and signal transmission. Understanding how proteins move is directly linked to understanding such processes, as well as to understanding dysfunctions and their contribution to diseases such as the mad cow disease and Alzheimer's disease [Selkoe 03].

In this thesis, we present a mechanistic modeling approach for proteins and show how it can be used to enhance molecular simulations. This modeling approach uses notions from robotics that allow high-level (coarse grained) treatment of molecules without loosing low-level (full-atom) details. We show how this modeling approach can be used to implement well-known and new Monte Carlo move classes as well as how it can lead to an overall enhanced sampling of the molecular conformational space. We also propose, based on this modeling approach, a combined motion planning and Normal Mode Analysis (NMA) [Cui 06] method for studying large amplitude motions in proteins. The use of the mechanistic modeling approach with the well-known RRT motion planning method [LaValle 01a] and normal mode analysis provides clear performance gains, which allow us to show results for the simulation of conformational transitions in proteins with up to one thousand residues.

In addition to the methodological contribution, this thesis also provides an extensive survey of the use of motion planning algorithms in molecular simulations. Up to our knowledge, the literature lacks such a survey, which would be useful for both roboticists and biologists willing to work in this domain.

The thesis is organized around these contributions as follows. Chapter 1 is dedicated to surveying and discussing the use of motion planning inspired methods in molecular simulations. Chapter 2 then presents the details of the mechanistic protein modeling approach, which acts as a basis for the methods presented in the proceeding two chapters.

Chapter 3 is dedicated to the applications of the modeling approach in Monte Carlo simulations. Next, Chapter 4 presents the combined RRT-NMA method and shows simulation studies for conformational transitions in proteins of various sizes. Finally, the thesis ends with a conclusion and a discussion of future research directions.

Chapter 1

Motion Planning Algorithms for Molecular Simulations

Motion planning is a fundamental problem in robotics that has motivated active research since more than three decades ago. A large variety of algorithms have been proposed to compute feasible motions of multi-body systems in constrained workspaces. In recent years, some of these algorithms have surpassed the frontiers of robotics, finding applications in other domains such as industrial manufacturing, computer animation and computational structural biology. This chapter concerns the latter domain, providing a survey on motion planning algorithms applied to molecular modeling and simulation. Both the algorithmic and application sides are discussed, as well as the different issues to be taken into consideration when extending robot motion planning algorithms to deal with molecules. From an algorithmic perspective, the chapter gives a general overview on the different extensions of sampling-based motion planners that have been proposed in this context. From the point of view of applications, the chapter deals with problems involving protein folding and conformational transitions, as well as protein-ligand interactions.

Since motion-planning-inspired algorithms for molecular simulations are relatively new, to our knowledge, no dedicated reviews have been written on this subject. Nevertheless, there are three works that are noteworthy in this regard. The first is a survey by Moll *et al.* [Moll 07] that is dedicated to applications of motion planning roadmap methods to protein folding only. The second is an online course prepared by Kavraki entitled "Geometric Methods in Structural Computational Biology"[Kavraki 07]. This course is a good and comprehensive reference on the broad subject of using geometric methods in computational biology. It is oriented towards explaining in detail the background, algorithms and the implementation details rather than surveying the current literature; which is the aim of this chapter. The third one is a very recent survey on computational models of protein kinematics and dynamics [Gipson 12]. This survey is focused on the

application of robotics-inspired methods together with Markov models to obtain a compact representation of the protein conformational space, which makes it limited in terms of the discussed methods and applications.

The aim of this chapter is twofold. First, it provides a basis for the next chapters by explaining concepts related to motion planning and how it can be used in molecular simulations. Second, it tries to fill the gap in the available literature by providing a comprehensive survey and discussion of the use of motion planning algorithms in molecular simulations. For readers in the structural biology community, this kind of survey can be looked as an introduction to robotics-inspired methods with applications in their domain, which will hopefully contribute to spreading the word about this new family of methods in the community. For readers in the robotics community, this kind of survey can incite them to look at problems in structural biology, which represent a challenging application domain that motivates the development of improved algorithms for accurate computations in very-high-dimensional spaces.

The chapter is organized as follows: Section 1.1 begins by introducing the general problem of motion planning and by presenting basic algorithms, especially sampling-based algorithms. The discussion then proceeds by explaining the different issues to be taken into account when moving from motion planning in robotics to performing molecular simulations. The main molecular simulation methods that are inspired by robot motion planning are then surveyed and explained in Section 1.2. Next, Section 1.3 discusses the three main application domains in computational structural biology where these algorithms have been applied. These application domains are: the analysis of conformational transitions, protein folding and unfolding, and protein-ligand interactions. For each of these domains, the general problem is presented and then results achieved using motion-planning-inspired techniques are surveyed and discussed. Finally, Section 1.4 summarizes and concludes the chapter.

1.1 From Robot Motion Planning to Molecular Simulations

This section introduces the motion planning problem and briefly presents some of the algorithms that have been proposed during the last three decades. More attention is given to the two classes of planning algorithms called Probabilistic Roadmap (PRM) [Kavraki 96] and Rapidly-Exploring Random Trees (RRT) [LaValle 01a], as robotics-inspired algorithms for molecular simulations mainly follow these approaches. The discussion will then proceed to how these algorithms can be extended for computing molecular motions.

1.1.1 Motion Planning in Robotics

The goal of robot motion planning is to decide automatically what motions a robot should execute in order to achieve a task specified by initial and goal spatial arrangements of physical objects [Latombe 90]. A frequently used example is: given a piano in a certain room, what motions should be applied to the piano in order to transfer it from position A to position B without colliding with any of the room's furniture? The formalized version of this problem is known as the Piano Mover's Problem [Schwartz 83].

Motion planning is generally formulated using the notion of Configuration Space [Lozano-Peréz 83]. A configuration q describes the pose of the robot (e.g. the x and y coordinates of a rigid robot translating in a 2D workspace). The configuration space C is the set of all possible configurations the robot can take, and the number of dimensions of this space equals the number of degrees of freedom of the robot (i.e. the number of parameters needed to describe the pose of the robot). Some regions in the configuration space may be considered forbidden due to the presence of obstacles or due to other constraints. These regions are usually denoted C_{obs} and the rest of the space is denoted C_{free}. The motion planning problem becomes a search problem in C_{free} for paths that connect the initial and goal configurations.

Early work focused on *complete* motion planning algorithms, i.e. algorithms that always report a solution if one exists and report failure otherwise [Goldberg 95, Latombe 90, LaValle 06]. An excellent overview of different classes of complete motion planning algorithms can be found in [Latombe 90] (Chapters 4 to 6). The problem with these methods is that they are inapplicable to problems with high dimensions or complex constraints. Finding complete solutions to such problems is known to be intractable [Reif 79, Canny 88]. For this reason, attention has shifted towards practical motion planning algorithms rather than complete ones. Sampling-based motion planners [Lindemann 05, Tsianos 07, LaValle 06] are such types of algorithms that have gained a lot of momentum lately. These algorithms trade off completeness for the sake of generality, efficiency and simplicity of implementation. They guarantee a weaker notion of completeness called *probabilistic completeness*, which means that with enough samples, the probability to find an existing solution converges to one [LaValle 06].

Sampling-based planners sample the configuration space to build a representative set of configurations instead of an explicit representation of the configuration space. Sampling-based planners are often classified into two categories: *roadmap-based* planners and *tree-based* planners. Roadmap methods work in two phases: a construction phase, where a graph that covers the configuration space is built, and a query phase, where the constructed graph is used to plan the motion between a start and goal configuration. These methods are also called multiple-query methods since the built roadmap can be used multiple times. Tree-based planners, on the other hand, are usually single-

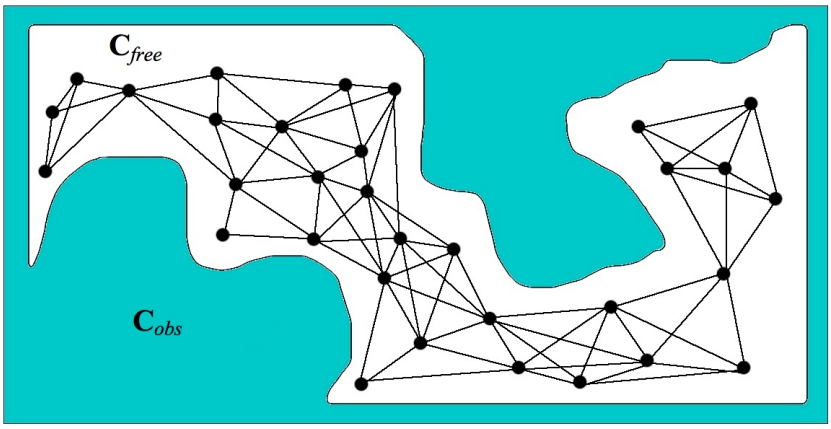

Figure 1.1: An illustration of a simple PRM.

shot methods. A tree is grown from the start configuration by sampling the space until a path to the goal configuration is found. Thus, the construction of the tree and the search for the path are done at the same time. The two algorithms described next, PRM [Kavraki 96, Geraerts 04] and RRT [LaValle 01b, LaValle 01a], are the most representative methods of each of these main classes. For more information about motion planning methods see [Canny 88, Latombe 90, Choset 05, LaValle 06].

Probabilistic Roadmap

The Probabilistic Roadmap (PRM) algorithm was introduced in the 1990s [Kavraki 96] and was able then to successfully solve motion planning problems with higher dimensions than what was achieved before. The basic version of PRM works by performing the following steps iteratively:

1. A random sample is drawn from the configuration space and is checked for collision. If the sample is a valid configuration, it is added to the roadmap as a node.

2. A search is performed to find the nearest neighbors in the roadmap to the new node.

3. An attempt is made to connect the new node to its neighbors using a local planner whose definition depends on the constraints imposed by the problem. If a connection can be established without collision, a new edge is added to the roadmap.

The roadmap is built by repeating the previous steps until a stopping criterion is met. Another version of the algorithm that performs sampling and connections in separate loops is also widely used. The produced graph can then be searched for paths using any of the conventional graph search algorithms such as Dijkstra's shortest path [Dijkstra 59] or the A* [Hart 72] algorithms. These basic steps of the PRM have been improved over the years and several variants have appeared (e.g. [Amato 98, Simeon 00, Wilmarth 02,

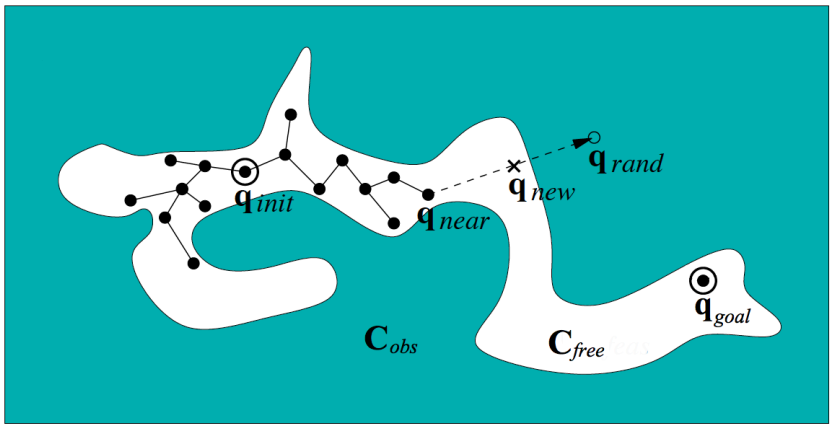

Figure 1.2: Illustration of a simple RRT at an intermediate stage during its construction.

Sánchez 03, Geraerts 04]). However, the general structure of the algorithm remains the same. Figure 1.1 shows an illustrative example of the basic PRM.

Rapidly Exploring Random Tree

The most popular tree-based motion planner is the Rapidly-exploring Random Tree (RRT) [LaValle 01b, LaValle 01a]. Rooted at the start configuration, a tree is iteratively constructed in the configuration space until the goal configuration can be connected to one of its nodes. An interesting feature of the algorithm is that nodes with larger Voronoi regions (i.e. the portion of the space that is closer to the node than to other nodes of the tree) are more likely to be chosen for expansion, and therefore the tree is pulled towards unexplored areas, spreading rapidly in the configuration space.

The basic version of the RRT works by performing the following steps iteratively:

1. A random configuration q_{rand} is sampled in the configuration space.
2. The tree is searched for a configuration q_{near}, which is the nearest node in the tree to q_{rand}.
3. A new configuration q_{new} is created by moving a predefined distance d from q_{near} in the direction of q_{rand} using a local planner or an interpolation method that depends on the mobile system.
4. If q_{new} is a valid configuration that falls in C_{free}, and if the local path between it and q_{near} is collision-free, then q_{new} is added to the tree as a new node and an edge is created between q_{new} and q_{near}.

This process is repeated until the goal configuration can be connected to the tree or a maximum number of iterations is reached. Figure 1.2 shows an illustrative example of the basic RRT algorithm. Variants of this basic algorithm appeared later on

(e.g. [Kuffner Jr 00, Bruce 02, Cheng 02, Rodriguez 06]). Moreover, other tree-based planners that are not directly based on RRT have also been proposed. Some examples of such planners are: Expansive Spaces Trees [Hsu 97], Path-Directed Subdivision Trees [Ladd 05] and KPIECE [Şucan 09].

1.1.2 Needed Extensions For Molecular Simulations

Since the algorithms discussed above have been developed with robotic applications in mind, they need to be extended or adapted in order to suit the requirements for simulating molecular motion. Generally speaking, there are several issues that need to be taken into account before applying such algorithms. First, a molecular representation that is suitable for applying motion planning algorithms needs to be adopted. Next, appropriate similarity measures (i.e. distance metrics) and collision detection methods for proteins need to be used. In addition, specific sampling methods can be required to satisfy structural constraints. Energies of molecular conformations also need to be taken into consideration since they determine the probability of their existence in reality. Furthermore, the very high dimensionality of problems involving biological macromolecules needs to be faced. These issues are discussed in the following along with a quick survey of the relevant literature.

Molecular Representation

The most straightforward way for representing molecules geometrically is to list the Cartesian coordinates of all the atoms [Leach 01, Koliński 10]. Bonds can then be constructed automatically using the distances between atoms and the knowledge about their types. This is called the *Cartesian representation* and it is used by the Protein Data Bank [Berman 02] to describe proteins. This representation is also frequent among conventional modeling tools based on Molecular Dynamics or Monte Carlo methods. The problem with such a representation is that it does not directly describe the internal degrees of freedom of the molecule.

There are three types of variables that can be considered as internal degrees of freedom in molecules: bond lengths, bond angles and dihedral angles. A bond length is the distance between two bonded atoms and a bond angle is the angle between two consecutive bonds. The dihedral angle around the bond between atoms A_{i-1} and A_i is the angle formed by planes A_{i-2}-A_{i-1}-A_i and A_{i-1}-A_i-A_{i+1}. See Figure 1.3 for an illustration. Although bond lengths and bond angles vary, their variation is known to be very small at room temperature [Schlick 10]. On the other hand, major conformational changes in the molecule occur due to variations in dihedral angles. For this reason, a widely adopted assumption is made, called the *rigid geometry assumption* [Scott 66], that considers dihedral angles to be the only degrees of freedom of the molecule. Hence, the conformation

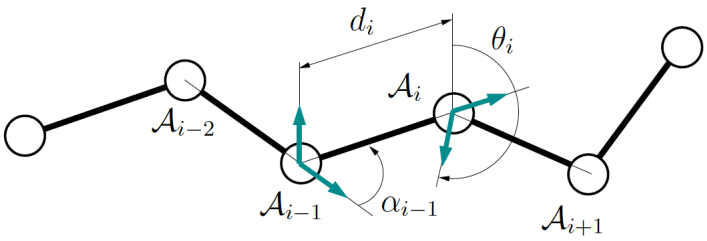

Figure 1.3: Parameters defining the relative position of bonded atoms.

of the molecule can be represented as a vector of only the dihedral angles [Leach 01]. This representation is called the *internal coordinates representation*. Figure 1.4 shows a protein model together with a representation of the dihedral angles corresponding to one of its amino acid residues.

Modeling a protein in internal coordinates is very similar to modeling an articulated robot. Indeed, modeling conventions applied in robotics can also be applied to molecules [Manocha 95, LaValle 00, Zhang 02, Noonan 05, Jagodzinski 07]. Based on the internal coordinates representation and the rigid geometry assumption, the protein can be looked at as an articulated mechanism, where bonds correspond to axes of revolute joints and atom-groups correspond to rigid links in a kinematic chain (for more about kinematic chains see: [Xie 03, Angeles 07, Sciavicco 01]). Finally it should be noted that the atom coordinates, which are required for some operations like energy computation and collision detection, can be computed from the internal coordinates using forward kinematics [Spong 06].

Dimensionality Reduction

Although using internal coordinates with the rigid geometry assumption reduces the number of variables, the number of degrees of freedom required to model biological macromolecules such as proteins remains very large. For example in molecular docking problems (see Section 1.3.3), ligands typically have 3-15 dihedral angles and receptors have in general more than 1000 dihedral angles, which makes the dimension of the combined search space prohibitively large [Teodoro 01]. This problem of high dimensionality is actually one of the major difficulties to be faced by computational methods in structural biology.

Several strategies have been used to reduce the dimensionality of the studied problems. For example, molecular docking problems have been tackled for a long time with the assumption that only the ligand is flexible and that the receptor protein is rigid [Leach 01]. However, since receptors may go through important conformational changes, it has been shown that this assumption leads to unrealistic solutions [Cavasotto 05b]. Other works

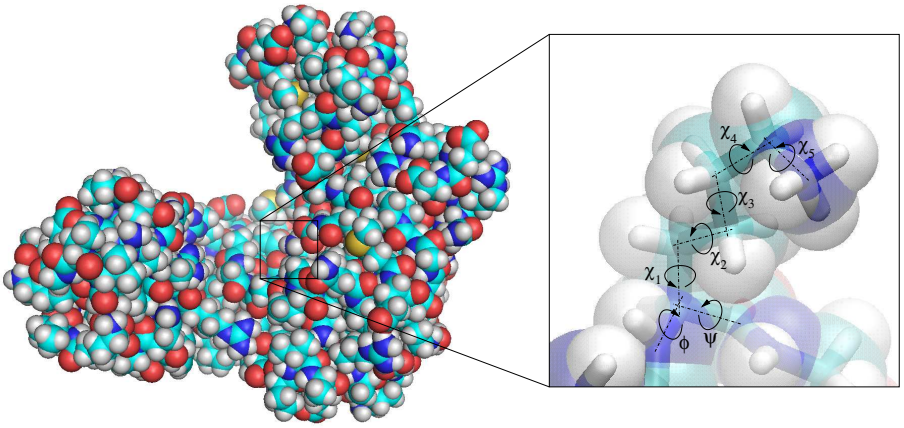

Figure 1.4: The main image shows a protein model in van der Waals representation (spherical atoms). The detail shows one of its constituent amino acid residues and the dihedral angles required to define its conformation.

(for e.g. [Jones 97a, Apostolakis 98, Pak 00]) have made more realistic assumptions based on prior chemical knowledge of the receptor protein. Using this knowledge, dihedral angles that contribute most to the motions of the receptor are identified. These dihedral angles are then assumed to be flexible and the rest of the receptor to be rigid. The drawback of such methods is that they are problem-dependent and hard to automate [Teodoro 01]. A more general approach proposed in [Thomas 07] identifies automatically which parts of the protein can be considered rigid using methods that are based on rigidity theory [Thorpe 99, Wells 05]. Another strategy to reduce the dimensionality of the problem is to assume that secondary structure elements are rigid, and that loops, linkers and side-chains are flexible. This approach, as in [Cortés 10b], reduces the number of variable parameters significantly and allows concentrating on important motions of the protein.

A different approach for addressing the problem is to use statistical dimensionality reduction methods [Fodor 02, van der Maaten 09] to map the current degrees of freedom into a lower-dimensional space. These methods usually begin with a previously-available ensemble of structures for the protein under study, which are analyzed in order to create a reduced set of degrees of freedom. An example of such methods is Principal Component Analysis (PCA) [Jolliffe 02], which is commonly used in the analysis of near-equilibrium fluctuations sampled by molecular dynamics simulations [Das 06, Altis 07, Mu 05]. In spite of the ability of PCA to capture important collective features, it may not be suitable for accurately representing large-amplitude molecular motions given that it provides a linear approximation and that molecular motions are generally non-linear. An example of methods that can capture non-linear features is the Isometric Feature Mapping (IsoMap) method [Tenenbaum 00]. This method produces a low dimensional space that preserves

as much as possible the geodesic distances between the conformations in the original high-dimensional space. This requires the construction of a nearest neighbor graph using a big number of distance computations, which makes the algorithm suffer when dealing with large datasets. A scalable version of IsoMap called Scalable IsoMap (ScIMAP) was introduced and applied to protein modeling applications [Das 06]. This method was further extended in [Plaku 07a] to be even more efficient by performing distance measures in yet another projection on a lower dimensional Euclidean space.

Normal Mode Analysis [Cui 06] has also been used in this regard. It has been shown that large-amplitude motions in proteins are related to low-frequency normal modes [Hinsen 98, Tama 01]. Consequently, low-frequency normal modes can be used to predict the direction of large-amplitude motions. In [Kirillova 08], transition pathways between conformations are computed using an RRT-like algorithm that explores linear combinations of low-frequency normal modes. An advantage of NMA over methods like PCA and IsoMap is that normal modes are computed from a single conformation, so that no dataset of conformations is required to be available *a priori*.

Distance Metrics

In molecular simulations, one often needs to measure how much a molecular conformation is different from or similar to another conformation. This notion of similarity (or distance) is also essential for most motion planning inspired methods. As explained in Section 1.1.1, RRT-based methods rely on finding the most similar conformation to every new random sample. PRMs also search for local connections between neighbor nodes corresponding to similar conformations. This makes the choice of the distance measure critical for the performance of the whole algorithm.

A widely used and straightforward distance measure is the coordinate root mean squared deviation (cRMSD), which is measured as the square root of the average squared distances between corresponding atoms in two molecules. This distance measure requires the conformations of both molecules to be aligned (superimposed) in order to remove the effect of any translation or rotation of the whole molecule. Examples of distance measures based on this idea include [Rao 73, Rossmann 76, Falicov 96]. Another widely used measure that eliminates the need to align the conformations is the distance root mean squared deviation (dRMSD). Here, distances are first computed between pairs of atoms of the same molecular conformation, then the root mean squared deviation is computed between these distances and the corresponding distances in the other molecular conformation. For an example of such a distance metric see [Holm 93].

Measuring the root mean squared deviation can also be done using dihedral values instead of atom coordinates, which is how robot configurations are typically compared within motion planning algorithms. Yet, it is important to note that in molecular sim-

ulations we are more interested in distance measures that capture structural differences in proportion to their effect on the potential energy of the molecule. Fluctuations in the backbone have generally a stronger effect on the energy than fluctuations in side-chains, for example. This is not the case with RMSD metrics in general, since they give the same weight to all-atom fluctuations regardless of how much these fluctuations affect the potential energy. For a comparison between different distance measures see [Wallin 03].

Computing distances can be a bottleneck for motion planning algorithms, especially if all-atom measures like dRMSD and cRMSD are used. Hence, several works have resorted to using approximate metrics instead of the exact ones. The rationale behind using such metrics is that an exact distance is not always required for the algorithm as a whole to function well, which justifies trading off exactness for the sake of performance gain. Several such methods can be found in the literature. One example is the work of Lotan and Schwarzer [Lotan 04], in which the protein is replaced by a lower dimensional *averaged* version that is used instead of the original one. This is done by subdividing the protein into n subsequences, each of which is replaced by its centroid. The authors used Haar Wavelet analysis to justify their metric and showed that it is highly correlated with the exact metric. Another example can be found in [Shehu 10]. In this work, the conformation of the whole protein is represented by only three variables that capture the overall topological differences between conformations. These variables are: the mean atomic distance to the centroid (ctd), the mean atomic distance to the farthest atom from the centroid (fct), and the mean atomic distance from the atom farthest from fct (ftf). An even more simplified metric is used in [Cortés 07] for the problem of molecular disassembly (see section 1.3.3), where the degrees of freedom of the protein side-chains and the torsions of the ligand are both ignored and only the reference frame associated with the ligand's geometric center is used for computing the distance.

A general method to devise simplified distance metrics, which could be applied for molecular simulations, is proposed in [Plaku 07b]. This method projects the sampled conformations q to an m-dimensional Euclidean space and performs the distance measures in that space. The projection is done by first selecting m pivots from q and then replacing each variable x_i in q by a vector of the distances between x_i and each of the pivots. Choosing pivots as far as possible from each other is believed to best preserve the distances as computed in the higher-dimensional space.

Collision Detection

Another important problem is the detection of collisions between parts of the same molecule and between different interacting molecules. As explained in Section 1.1.1, sampling-based algorithms need a collision checker to decide at every step if a new conformation is valid, and to check if two adjacent conformations can be connected by a

collision-free path. Collision detection is indeed intensively performed inside these algorithms. Very efficient collision checkers tailored for molecular models are therefore necessary for the overall efficiency of the planning algorithms.

Collision detection has been widely studied in the fields of robotics and computer graphics [Jiménez 01, Lin 03] and several general-purpose collision detection packages are available (e.g. [Gottschalk 96, van den Bergen 98, Cohen 95]). However, the problem with most of these methods is that they do not directly address the complex chain-like structure of large molecules such as proteins. This makes such methods less efficient than what can possibly be achieved, since the number of pairs considered for collision in the chain can be significantly reduced by exploiting the structural properties of the chain (see [Soss 03, Agarwal 04] for some examples of works that address the specific problem of collision detection in kinematic chains).

Several algorithms dedicated to chain-like molecular models have been proposed. The technique described in [Lotan 02] exploits the topology of the molecular (kinematic) chain to avoid testing for self-collision parts that are known to be rigid. It uses a hierarchical representation of the chain that allows for efficient updates and queries in $O(logN)$ time, and superimposes on top of this representation a hierarchy of bounding boxes, which allows for efficient collision detection and distance computation. The algorithm detects self-collisions with a worst-case complexity of $O(N^{4/3})$. Another algorithm, called BioCD [de Angulo 05], was specifically designed to be used within sampling-based motion planning algorithms applied to proteins described as kinematic chains. It assumes that only a pre-selected set of the degrees of freedom of the protein can change arbitrarily and the rest are blocked. The algorithm works by creating a two-level hierarchy that allows it to avoid detecting collisions between atom pairs whose distance does not change from one iteration to another.

Loop Closure

Loops are portions of proteins that are highly irregular and varied in terms of their sequence and structure. They can play important roles in controlling enzyme activity, and are often found at the interface in protein-protein or protein-DNA/RNA complexes [Rangwala 10]. Sampling such portions of the protein poses a challenge that requires extra care. Conformations of loops must not only satisfy geometric constraints for collision avoidance, but must also satisfy what is known as the *loop-closure* constraint. The two ends of the loop must remain bonded to the rest of the molecule, which greatly restricts the space of admissible conformations of the molecular chain. Therefore, defining an appropriate sampling strategy is a prerequisite for any sampling-based exploration method that takes loop flexibility into consideration.

The protein loop closure problem has often been addressed using robotics-inspired

methods (e.g. [Coutsias 04, Kolodny 05]). Note however that most such methods are limited to 6 degrees of freedom, and therefore, extensions are necessary to deal with long loops. In [Cortés 05a], an algorithm called *RLG* (short for Random Loop Generator) was proposed for sampling configurations of long loops. The main idea of RLG is to decompose the loop into several parts: a *passive chain* and one or two *active chains*. RLG progressively constructs a random configuration for the active chains by alternating sampling between them. This sampling is performed in a way that increases the probability of satisfying loop closure when finding a configuration for the passive chain, which is computed by solving inverse kinematics for 6 consecutive bond torsions. In [Cortés 05b], a modification was introduced to RLG for enhancing its efficiency. The idea was to include steric-clash checks during the sampling of the active chains, rather than only after the complete conformation is generated. In [Yao 08], another sampling strategy for protein loops is proposed that works in a similar manner to RLG. It decomposes the loop into three parts called: front-end F, mid-portion M and back-end B, samples F and B first, and then uses inverse kinematics to find a conformation for M.

An alternative to the methods above, which apply (semi-)analytical inverse kinematics, is to use optimization-based inverse kinematics. Examples of such methods include the Cyclic Coordinate Descent (CCD) [Canutescu 03] and the method introduced in [Lee 05].

Energy Computation

As mentioned in Section 1.1.2, there is a high similarity between the representation of robot configurations and molecular conformations. Yet, there is a fundamental difference that needs to be taken into account whenever dealing with molecules, which is the potential energy associated to conformations. Each molecular conformation has an energy level that depends on the interactions between its constituent atoms and with the surrounding molecules (e.g. the solvent). This energy is an indicator of how likely it is for the molecule to adopt this conformation (conformations with low energy are naturally preferred over conformations with high energy). Hence, the conformational space of the protein is not a binary space with only valid or invalid conformations, but a continuous space with conformations that are more or less likely to occur. For many applications, the algorithms must be able to find *least energy paths* rather than geometrically valid ones. Therefore, sampling-based algorithms need to be adjusted to cope with this by accepting or rejecting new conformations based on their energy level, and by associating transition probabilities between conformations based on the energy difference between them.

The energy of a conformation can be computed with high precision using quantum mechanics [Griffiths 05]; however, it is highly time consuming and can be even intractable in large molecules, since it deals directly with the electronic structure of the molecule. Molecular mechanics [Burkert 82] is usually used to provide approximate energy values of

protein conformations. Functions that compute energy based on molecular mechanics are usually called *molecular force fields*. They take as input the atom positions and evaluate energy based on different terms that vary from one force field to another. Yet, these terms usually include: changes in bond lengths and bond angles, bond torsions, van der Waals interactions and electrostatic interactions. The choice of the terms and the shape of the function affect the accuracy of the computation, its speed, and its suitability to some types of molecular systems or applications. See [Ponder 03, Mackerell Jr 04] for reviews on force fields and software packages that are widely used in the study of proteins.

The drawback of using such all-atom force fields is that they are still computationally expensive, and thus their usage can limit the size of the studied molecules and the time-scale of the performed simulations. This has motivated the introduction of *coarse-grained* force fields [Tozzini 05]. These force fields measure interactions between blocks of functional groups rather than between the individual atoms. This leads to a rough approximation of the actual force field, but also to a significant performance gain. Some examples of coarse-grained force fields are MARTINI [Monticelli 08] and OPEP [Derreumaux 99].

1.2 Motion Planning Inspired Methods for Molecular Simulations

A seminal work on the application of motion planning algorithms to the study of proteins was published in 1999 [Singh 99]. Since that time, many methods inspired by different motion planning algorithms have appeared and have been applied to a variety of molecular simulation problems. Most of these methods follow the lines of either PRM or RRT, with PRM-based methods being more oriented towards the computation of ensemble properties and RRT-based methods more towards the computation of feasible paths. In this section, we survey literature related to these methods and provide brief explanations of each of them.

1.2.1 PRM-Based Methods

Probabilistic Conformational Roadmaps

The method proposed by Singh *et al.* [Singh 99] builds a roadmap by randomly sampling the molecular conformation space. Samples are accepted or rejected using a probability function that favors low energy conformations. This feature makes the method different from the conventional PRM in robotics that uses collision detection for evaluating new

samples. The probability function used is as follows:

$$P_{accept}(q) = \begin{cases} 1 & if\ E_q < E_{min} \\ \frac{E_{max} - E_q}{E_{max} - E_{min}} & if\ E_{min} \leq E_q \leq E_{max} \\ 0 & if\ E_q > E_{max} \end{cases} \tag{1.1}$$

where E_q is the potential energy of conformation q, and E_{min} and E_{max} are threshold values that depend on the molecular system in hand. Neighboring nodes are then connected, and a weight is associated to each edge. These weights are probabilities that represent the likelihood of transitions between the connected conformations. For each edge e_{ij}, the algorithm generates intermediate conformations $\{q_i = c_0, c_1, c_2, ..., c_n = q_j\}$ along the path between the two connected conformations q_i and q_j. The number of these intermediate conformations is a user-defined parameter. The weight of the edge e_{ij} is then computed by summing the negative logarithm of the transition probabilities between each of the consecutive intermediate conformations c_i and c_{i+1}:

$$P_i = \frac{e^{-(E_{i+1} - E_i)/KT}}{e^{-(E_{i+1} - E_i)/KT} + e^{-(E_{i-1} - E_i)/KT}} \tag{1.2}$$

where E_i is the energy of c_i, T is the temperature and K is the Boltzmann constant. A connectivity-enhancement step is also added to this PRM variant, by sampling extra nodes around nodes that have very few edges.

This method was first introduced for the study of protein-ligand interactions, more precisely, to identify potential active sites in the proteins. The weights of paths entering and leaving low energy nodes were also used to estimate energy barriers around active sites and to distinguish true binding sites from other low-energy active sites. Later, in [Apaydin 01], this method was given the name of Probabilistic Conformational Roadmaps (PCR), and was applied to study protein folding.

Stochastic Roadmap Simulations

Stochastic Roadmap Simulations (SRS) [Apaydin 02, Apaydin 03, Apaydin 04, Chiang 06, Chiang 07] is an evolution of PCR. The main difference between the two methods is found in the transition probability assigned to edges in the roadmap. SRS uses a transition probability that is consistent with the Metropolis criterion [Metropolis 53, Frenkel 02], which allows establishing a connection between SRS and Monte Carlo methods. The transition probability used in SRS is as follows:

$$P_{ij} = \begin{cases} \frac{1}{n_i} \exp(-\frac{\triangle E_{ij}}{KT}) & if\ \triangle E_{ij} > 0 \\ \frac{1}{n_i} & otherwise \end{cases} \tag{1.3}$$

$$P_{ii} = 1 - \sum_{j \neq i} P_{ij} \tag{1.4}$$

where $\triangle E_{ij}$ is the difference in potential energy between nodes q_i and q_j, and n_i is the number of neighbors to q_i. As in equation 1.2, T is the temperature and K is the Boltzmann constant. A self-transition edge is added to each node such that the sum of transition probabilities for every node is one.

Once the roadmap is constructed, tools from Markov Chain Theory (e.g. First Step Analysis) can be applied to study ensemble properties like folding rates, phi-values and the Transition State Ensemble (see Section 1.3.2). Every path in the roadmap can be considered as a run of the Markov Chain Monte Carlo (MCMC) method. This allows interpreting the whole roadmap as the result of a set of MCMC explorations being run simultaneously. In fact in [Apaydin 03], SRS is shown to converge at the limit to the same sampling distribution as that of MCMC. The difference between MCMC and SRS is that MCMC provides a single but fine-grained path, whereas SRS provides many coarse-grained paths covering a wider area of the conformational space. This is of course a tradeoff, since although SRS covers a wider area of the space in a relatively short time and overcomes the local minima problem inherent to MCMC, coarse granularity comes at the cost of possibly losing important information along the paths between nodes.

PRMs for Folding Pathways

Another early research direction is the work led by Nancy Amato [Song 02, Song 03, Amato 03, Thomas 05, Tang 05, Tapia 07, Thomas 07, Tang 08, Tapia 10]. The PRM-based algorithms proposed by this group to study protein (un-)folding are largely inspired by the PCR method. The method builds a roadmap by sampling the conformational space of the protein with a probability function that is similar to that of PCR (see equation 1.1). New samples are first checked for collisions between atoms and then accepted or rejected based on the probability function. In this function, E_{min} is suggested to be set to the potential energy of the extended chain and E_{max} to be twice E_{min} [Tapia 10]. This method also assigns weights to edges in order to find the most likely paths. The equation to compute these weights is exactly the same as the one used to determine the move acceptance probability in Monte Carlo methods, usually called the Metropolis criterion [Metropolis 53, Frenkel 02]:

$$P_i = \begin{cases} e^{\frac{-\triangle E_i}{KT}} & if \ \triangle E_i > 0 \\ 1 & otherwise \end{cases} \tag{1.5}$$

where $\triangle E_i = E(c_{i+1}) - E(c_i)$, T is the temperature and K is the Boltzmann constant.

This method has gone through several evolutions over time. Changes mainly con-

cern the strategy used for sampling new nodes and the method used to analyze folding pathways. The three main sampling strategies are summarized in the following:

1. In [Song 02, Song 03], sampling was performed around the native fold (which is assumed to be known) using a set of normal distributions centered around this conformation with various standard deviations. This was done to ensure capturing important details close to the native fold using small standard deviations and to ensure adequate coverage of the conformational space using larger standard deviations.

2. In [Amato 03, Thomas 05] another strategy was proposed since the previous one worked well only for proteins containing up to 60 residues. The new strategy also starts from the native fold but generates new conformations by iteratively applying small perturbations. Conformations are partitioned into bins according to the number of native contacts present. A native contact is defined as a pair of C_α atoms that are within 7 Å of each other in the native state. At each round, bins with a small number of conformations are chosen and sampling is performed around them. Newly generated conformations are placed at the appropriate bins and the loop repeats.

3. The last method based on native contacts was also found to scale poorly beyond proteins with 100 residues. In [Thomas 07], another totally different method was proposed for sampling based on *rigidity analysis*. Here, the protein is analyzed to identify three types of bonds: rigid bonds, flexible bonds whose motion does not affect other bonds (called *independently flexible*) and flexible bonds that form a set such that the motion of any of them affects the rest of the set (called *dependently flexible*). The method perturbs rigid bonds with a low probability denoted P_{rigid} and independently flexible bonds with a high probability denoted P_{flex}. For each set of dependently flexible bonds, a number of bonds are chosen randomly and are perturbed with probability P_{flex}, whereas the others are perturbed with probability P_{rigid}. This method was able to characterize the energy landscape more efficiently, with fewer and more realistic conformations.

Works derived from this method have been proposed more recently by other researchers. An example is the MaxFlux-PRM [Yang 07, Li 08], which uses a slightly different edge weight function in order to find temperature-dependent optimal reaction paths. In this algorithm, edge weights are computed as a function of the exponential variation of the energy and the distance between conformations.

1.2.2 RRT-Based Methods

Basic RRT variants for computing molecular motions

The first works on the application of RRT to molecular simulations [Cortés 04, Cortés 05b] were based on a basic variant of the algorithm. The referred papers present a two-stage approach. In the first stage, RRT is applied on a mechanistic representation of the molecular system, only considering geometric constraints. Paths resulting from the first stage are then analyzed and refined in a second stage using a more accurate energy model. The advantage of this two-stage approach is that large-amplitude motions can be computed with few computational resources. The performance of the method was investigated on several classes of problems involving protein loop motions and protein-ligand interactions.

A similar approach was proposed in [Enosh 08] for the simulation of conformational transitions of proteins. The main difference with the aforementioned method concerns the validity test performed during the RRT construction, which includes an energy evaluation in addition to the geometric constraints. The authors also proposed a method to cluster paths computed from several runs of RRT in order to facilitate the analysis performed in a second stage. The technique, based on path alignment, was also used to compute the most energetically favorable path in the solution set by combining portions of different solutions.

An improvement of the aforementioned RRT-based method, called PathRover, was proposed in [Raveh 09]. In this work, a *branch-termination* scheme is applied to limit the exploration to a subset of the conformational space that satisfies a set of constraints based on prior information. This scheme works by representing partial information from previous experiments and expert knowledge as predicates that are checked periodically as the RRT grows. Branches of the tree that do not improve a certain predicate after m consecutive iterations are terminated (not extended anymore).

Manhattan-Like RRT: Decoupling degrees of freedom

The Manhattan-like RRT (ML-RRT) algorithm proposed in [Cortés 08] was developed to circumvent the limitations of the basic RRT algorithm to deal with high-dimensional problems in the particular context of *(dis)assembly path planning*. This is a variant of the motion planning problem that consists of finding a path to (dis)assemble two objects, one of which is considered to be mobile, and the other one to be fixed. In the more general instance addressed here, both the mobile and the fixed object contain articulated parts. This problem resembles the problem of computing access/exit paths for a ligand (small molecule) to/from the active site of a protein (see Figure 1.5 for an illustration).

The main idea of ML-RRT is to divide configuration/conformation parameters into

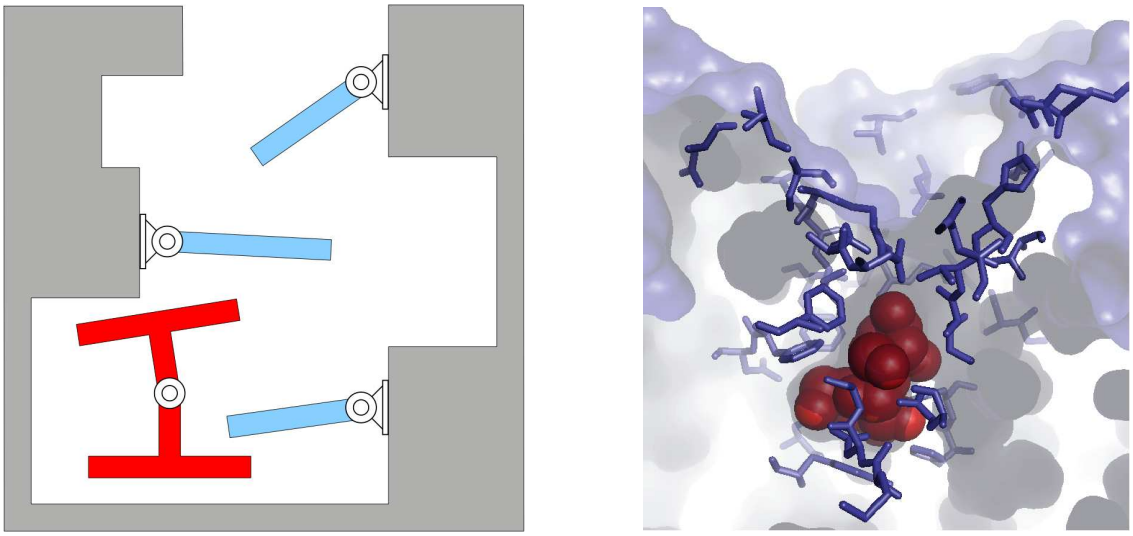

Figure 1.5: The image on the left illustrates an academic disassembly planning problem for two articulated objects. An analogy can be made with the protein-ligand "disassembly" problem represented in the right-hand image. The red object can be considered as the ligand and the blue sticks as flexible side-chains of the protein.

two groups, called active and passive, and to generate their motion in a decoupled manner. Active parameters correspond to parts whose motions are essential for the disassembly task, whereas passive parameters correspond to parts that need to move only if they hinder the motions of other mobile parts (active or passive). Roughly speaking, motions of active parts are planned exactly the same way they are planned using RRT, but when motion is hindered by a passive part, the conformation of this part is perturbed in order to allocate free space for the motion of active parts. The performed perturbation may also cause collisions with other passive parts, which are then perturbed producing a domino-like effect.

The ML-RRT algorithm presents two main advantages when compared to the basic RRT. First, it is considerably faster, and second, it allows identifying automatically (without user intervention or the need of prior knowledge) which parts of the protein need to move in order for the ligand to enter or exit from the active site.

The original ML-RRT algorithm is able to solve efficiently problems involving the flexibility of the ligand and the protein side chains. The extensions proposed in [Cortés 10b] enable the introduction of the protein backbone flexibility. In this extension, the protein is represented as groups of rigid bodies connected by flexible loops that are assigned based on structural knowledge. Additionally, a mobility coefficient is assigned to each passive parameter. This coefficient is used to differentiate passive parts that are allowed to move easily from those that should be moved only if the solution path cannot be found otherwise.

Transition-RRT: Exploring energy landscapes

Another RRT variant called Transition-RRT (T-RRT) was introduced in [Jaillet 10, Jaillet 11] for exploring energy landscapes. The algorithm introduces a state transition test inspired from the Metropolis criterion in MC methods. The goal is to favor the exploration of low-energy regions. New nodes are accepted and added to the tree with a probability given by equation 1.5. In this equation, $\triangle E_i$ is the difference between the energy at the new candidate node (q_{new}) and its nearest neighbor in the tree (q_{near}). In contrast to MC methods, where the temperature T is usually a constant for the simulation, T-RRT incorporates a reactive scheme to dynamically adapt this parameter. To do so, the algorithm keeps track of the number of consecutive tree expansion rejections. When the T-RRT search reaches a maximum number of consecutive rejections, the value of T is increased, which increases the probability to accept subsequent transition tests. In contrast, each time an uphill transition test succeeds, the value of T decreases, therefore increasing the severity of the transition test. Thus, the temperature is automatically regulated during the exploration depending on the shape of the energy landscape. This temperature regulation strategy is a way to balance the search between unexplored regions and low energy regions. Note that T-RRT does not yield a Boltzmann-weighted set of conformations. However, it allows finding efficiently energy minima and saddle points in the energy landscape, as well as likely transition paths between stable conformations.

Recently, the underlying principles of ML-RRT and T-RRT have been combined within an algorithm called MLT-RRT [Iehl 12]. The combined approach extends the practical applicability of T-RRT to higher-dimensional problems in which the energy (or cost) function can be decomposed as a sum of elementary terms associated with subsets of configuration/conformation parameters.

NMA-RRT: Exploring collective motions

The work by Kirillova *et al.* [Kirillova 08] proposes an RRT-based method that applies Normal Mode Analysis (NMA) [Cui 06] for computing global macromolecular motions. As mentioned in Section 1.1.2, low-frequency normal modes are associated with collective, large-amplitude molecular motions, and can be used as predictors for the direction of such motions. This property is exploited by the NMA-RRT method, which performs an RRT-like exploration in the coordinate space of the low-frequency normal modes. The goal is to cover the most important areas of the conformational space while exploring a low-dimensional search space. Although NMA-RRT performs its search in a space that is defined in terms of the amplitudes of low-frequency normal modes and not in terms of the degrees of freedom of the molecular model, new conformations are accepted only if they satisfy the geometric constraints of the mechanistic model (i.e. correct bond geometry, collision avoidance). Normal mode calculations are iteratively updated during

the conformational search. This is necessary because the information provided by NMA is only accurate in a relatively small region around the initial conformation, which causes the guidance of the RRT search to degrade when exploring larger regions.

1.2.3 Other Methods

In addition to the aforementioned methods, several methods for molecular modeling and simulation that apply ideas from motion planning algorithms other than PRM and RRT have been proposed in recent years.

In [Shehu 10], Shehu *et al.* proposed a tree-based method called Fragment Monte Carlo Tree Exploration (*FeLTr*) for protein structure prediction (see Section 1.3.2). This method grows a tree in the conformational space that tries to guide the search toward low-energy regions while avoiding oversampling geometrically similar conformations. The tree is expanded with low-energy conformations through a fragment-based Monte Carlo sampling strategy. The goal of FeLTr is to locate low energy conformations that are potentially close to the protein's native conformation. These native-like conformations can then act as starting points for a more refined search to obtain the folded conformation.

A similar two-step approach for protein structure prediction, called Model Based Search (MBS), is described in [Brunette 08]. MBS starts by running short MC simulations with a coarse-grained energy model. A tree-based clustering algorithm is then used to group the sampled conformations into funnels that represent coherent regions in the conformational space. Full-atom energy evaluation using Rosetta [Rohl 04] is then used to identify relevant funnels that are further explored with refined MC runs.

Another motion-planning-based method was introduced in [Haspel 10] for computing large-amplitude motions between molecular conformations. The method is based on the Path Directed Subdivision Tree (PDST) algorithm [Ladd 05], which is also a tree-based sampling-based planner, but which represents samples as path segments rather than individual states, and uses non-uniform subdivisions of the space to estimate coverage [Ladd 05]. The space subdivision is based on a distance metric defined in terms of the relative positions between the secondary structure elements. In order to enhance the performance of the method, a coarse-grained protein model and a simplified energy function were considered.

1.3 Applications

The methods presented in the previous section have been mainly applied to three types of problems in computational structural biology: the simulation of conformational transitions of proteins, the study of the protein folding process, and the analysis of protein-ligand interactions. This section discusses briefly each of these problems and presents the

main results achieved by motion planning inspired methods.

1.3.1 Conformational Transitions

The most direct application of robot motion planning methods in molecular simulations is the computation of transition pathways between two molecular conformations. This problem requires generating a sequence of feasible intermediate conformations for the molecule (usually a protein) to link two given states. The problem is analogous to the motion planning problem in robotics. This problem can be seen as a general instance of several more specific problems. In *protein folding* for example, the starting and end conformations are the unfolded and folded states of the protein, and in *molecular docking*, the starting and end conformations are the undocked and docked states of the molecular complex. These two particular problems are treated in the next subsections. This section concerns transitions between stable (folded) states of proteins.

The study of protein conformational transitions is important since they can play key roles in molecular recognition and may be essential for the protein activity. In spite of their importance, current experimental and computational methods are very limited for describing large-amplitude conformational changes in proteins at the atomic scale.

Finding transition pathways is usually tackled at different levels of granularity depending on the studied problem. Some studies are related to large-amplitude motions that occur over a relatively long period of time and that significantly affect the whole protein (such motions are often referred to as domain motions). In such cases, the problem can be tackled at a structural level, with lower resolution than the atom level. In other cases, interest may be focused on flexible segments of the protein. For example, irregular segments, called loops and linkers, are generally much more flexible than structured parts of the protein (i.e. alpha helices and beta sheets). This calls for exploration methods that are specifically tailored for these flexible regions. Figure 1.6 illustrates these two types of protein motions.

Loop Motions

The first application of an RRT-based algorithm extended to treat closed kinematic chains (RLG-RRT) [Cortés 05a] for computing protein loop motions was described in [Cortés 04]. The algorithm was applied to study the mobility of loop 7 in *amylosucrase* (AS). This is a long loop involving 17 amino acid residues. The articulated closed-chain model of the loop contains 51 degrees of freedom. Results showed a possible opening/closing motion of this loop (similar to that of other enzymes), and served to demonstrate the effectiveness of motion-planning-based methods for studying the mobility of protein loops. An improved version of the method, which integrates ideas of ML-RRT, was applied in [Barbe 11] to investigate the large-scale open-to-closed movement of the lid that controls

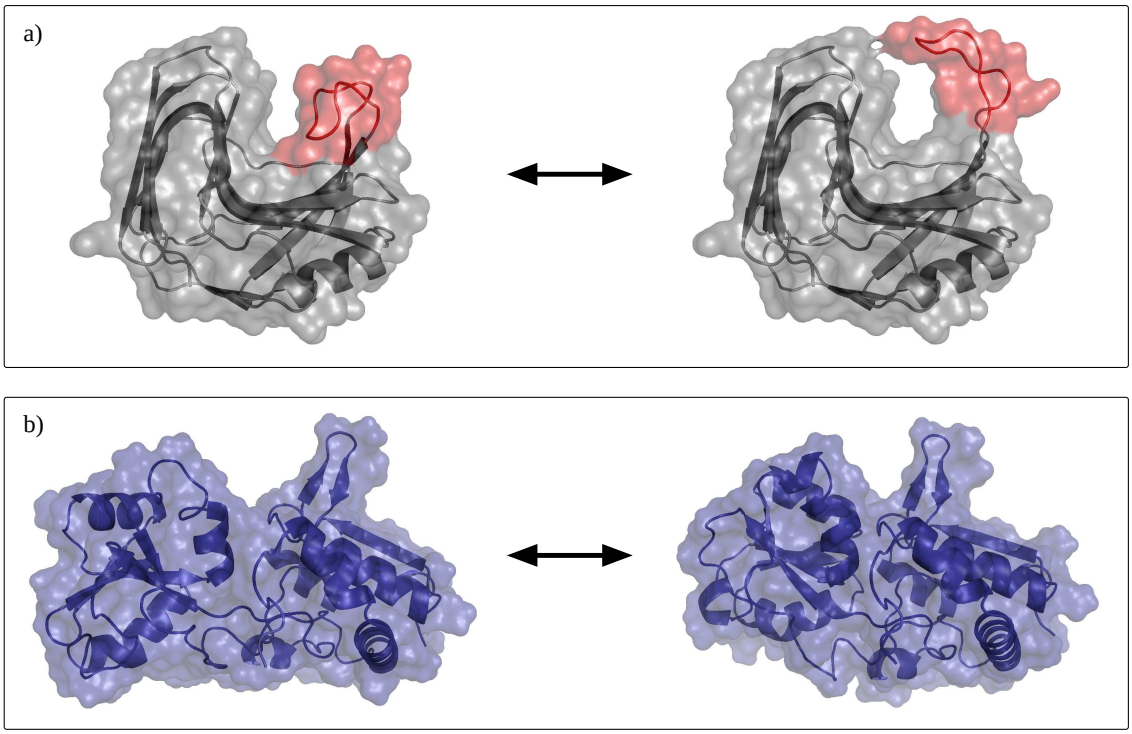

Figure 1.6: Illustration of two classes of large-amplitude motions in proteins. (a) Loop motions: a segment of the protein (in red) moves significantly, while the rest of the protein remains mostly static. (b) Domain motions: large portions of the protein move with respect to each other.

the access to the active site of *Burkholderia cepacia lipase* (BCL). Results showed that the lid conformational transition computed with this method is comparable to the one obtained with molecular dynamics simulations. Nevertheless, the computing time required by the RRT-based method is several orders of magnitude lower (a few hours on a single processor compared to weeks on a medium-sized cluster).

Several tests on the application of LoopTK to study motions for 20 different loops are presented in [Yao 08]. Results show that LoopTK can sample efficiently conformations of loops ranging from 5 to 25 residues in length. Although the combination of LoopTK with sampling-based path-planning algorithms such as PRM and RRT seems possible, results on the application of such a combined strategy to simulate protein loop motions have not been published yet, as far as we know.

Domain Motions

The results reported in [Kirillova 08] show the good performance of NMA-RRT for computing transition pathways involving domain motions. A set of five proteins for which structures corresponding to different conformations have been experimentally solved was

used as a benchmark. The abbreviated names of these proteins are: ADK, ATP, DAP, EIA and LAO. Further tests on *adenylate kinase* (ADK) showed that NMA-RRT produces results that correlate well with previous studies [Maragakis 05]. Remarkably, NMA-RRT was able to achieve these results using a very low number of normal mode calculations.

Results obtained with PathRover for computing conformational transitions of the *CesT* and the *Cyanovirin-N* proteins are reported in [Raveh 09]. The particular phenomenon studied in these tests is *domain swapping*, and the achieved results were consistent with experimental results. Moreover, in [Enosh 08], the RRT-based predecessor of PathRover was implemented within a larger framework of algorithms to generate pathways between a closed and an open conformation of the *KcsA* protein, providing interesting insights into this process.

Conformational transition simulations have also been performed using the PDST-based method presented in [Haspel 10] (see Section 1.2.3). Results are reported for the *ADK*, *RBP*, *GroEL* and *CVN* proteins. These results show that the algorithm significantly outperforms a classically used method such as Simulated Annealing [Kirkpatrick 83]. The paper also shows that results of the PDST-based method are consistent with experimental data.

1.3.2 Protein Folding

Protein folding is the process in which proteins move (fold) from random coils to their native three-dimensional shape. For an illustration, Figure 1.7 represents folded and unfolded conformations of a small protein. Being in the correct folded state is essential for proteins to function properly, and, usually, unfolded or incorrectly folded proteins are inactive or even toxic [Dobson 03, Selkoe 03]. For this reason, it is important to understand and to characterize protein folding and unfolding pathways. Note that the study of protein folding should be distinguished from the problem of protein structure prediction [Zaki 08], in which only the final three-dimensional structure of the protein is searched, regardless of how the protein actually reaches it. Nevertheless, both problems are important, and progress in any of them may yield advances in the other.

Several experimental methods have been used for studying protein folding, such as NMR Spectroscopy [Balbach 95, Dyson 04], Ultrarapid Mixing [Chan 97] and Time-Resolved Absorption Spectroscopy [Jones 93]. However, these methods are currently limited in their ability to capture short-lived events and to characterize conformations with a high spatial resolution. Computational methods have been used side by side with these experimental methods, either augmenting them or even replacing them (for examples, see [Unger 93, Sugita 99, Onuchic 04, Dill 08]). Important advances with these computational methods started with the advent of the energy landscape theory [Bryngelson 95], which hypothesizes that the energy landscape of a protein is funneled with many path-

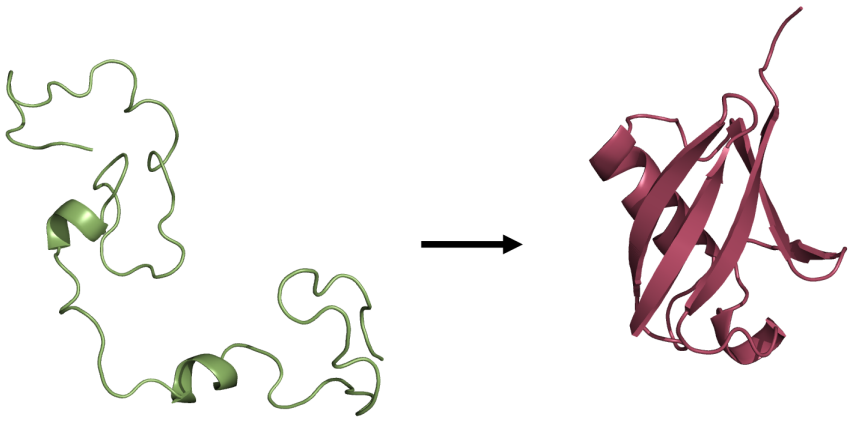

Figure 1.7: A small protein (*ubiquitin*) in an unfolded state (left) and folded state (right).

ways all leading to the same final folded state. This suggests that a good understanding and characterization of the energy landscape of a protein will lead to a good understanding of how this protein folds. Hence, motion planning inspired methods for protein folding basically take this theory as a basis. The advantage of such methods over most conventional methods is their ability to rapidly explore the conformational space without getting trapped in local energy minima, and their capacity to find several pathways simultaneously.

Computation of Folding Quantifiers

There are several types of quantifiers that are used for studying and expressing properties of protein folding pathways. These quantifiers can be computed using experimental methods, which makes them useful also for evaluating the performance of computational methods. Examples of the most frequently used quantifiers are:

- The probability of folding (P_{fold}), which is the probability that the structure at a certain conformation will become completely folded before it becomes completely unfolded.

- The Transition State Ensemble (TSE), which is the set of conformations with $P_{fold} = 0.5$ (i.e. conformations which make up the energy barrier the protein must cross in order to fold).

- The folding rate, which corresponds to an experimentally measurable quantity that determines how fast a protein proceeds from the unfolded state to the native folded conformation.

- The Φ-value, which measures how close a certain residue is to its native folded state.

In [Apaydin 03, Apaydin 04], P_{fold} values were computed and compared using SRS and Monte Carlo (MC) for two proteins with PDB IDs 1ROP and 1HDD. These proteins were modeled at the secondary structure level with 6 and 12 degrees of freedom respectively. Results showed that SRS computations improve rapidly as the roadmap size increases, and that the correlation between SRS and MC computations tends to increase as more MC runs are performed per node. Nevertheless, SRS produced results at least four times faster than MC. More extensive tests were presented in [Chiang 06, Chiang 07], where 16 proteins were analyzed using SRS to compute TSEs, folding rates and Φ-values. Results were then compared to those obtained with an existing dynamic programming method and were found to better estimate experimental data when computing TSEs and folding rates. However, both SRS and the dynamic programming method did not produce very good estimates for Φ-values.

PRM-based methods have also been applied to compute folding quantifiers together with two new analysis methods called *Map-based Master Equation* (MME) and *Map-based Monte Carlo* (MMC). These methods were introduced in [Tapia 07] and used in combination with the conformational exploration method presented in Section 1.2.1 to compute relative folding rates for proteins *G*, *NuG1* and *NuG2*. These analysis methods are extensions to the original Master Equation and Monte Carlo techniques, and they are applied on the constructed roadmap instead of the full conformational space as is conventionally done. The computed relative folding rates were found to match the corresponding experimental data.

Finally, the capacity of FeLTr to predict native-like conformations of small-to-medium size proteins has been shown in [Shehu 10]. Results in this paper show a good performance of the method on eight proteins, modeled with 40 to 152 degrees of freedom. The conformations provided by FeLTr can be used as starting points for more detailed biophysical studies.

Protein (Un)folding Pathways

Results on the performance of PRM-based methods for studying unfolding of several proteins with up to 100 residues are reported in [Song 02, Song 03, Amato 03, Thomas 05]. The constructed roadmaps were used to extract unfolding pathways and to identify their *secondary structure formation order*. The results were found to be in good agreement with known experimental data. This method was tested on the proteins *G* and *L*, as well as on proteins *NuG1* and *NuG2*, which are two mutants of protein *G*. Initial tests in [Song 03] were able to capture the folding differences between proteins *G* and *L*, but not between *G* and *NuG1* or *NuG2*. However, these differences were correctly captured after applying the rigidity-based sampling strategy in [Thomas 07].

RNA (Un)folding Pathways

The combination of the PRM-based exploration with MME and MMC discussed above has also been used in [Tang 05, Tang 08] to study the problem of RNA (un)folding, which is a problem that is very similar to protein folding. Results show that the method scales well for RNA molecules with up to 200 nucleotides. This method was used to compute relative folding rates, and was found to agree with experimental results. It was also able to predict the same relative gene expression rate for wild-type MS2 phage RNA and three of its mutants.

1.3.3 Protein-Ligand Interactions

The study of protein-ligand interactions is essential for understanding many biological mechanisms. In terms of applications, understanding such molecular interactions is essential for drug design in pharmacology, or for protein engineering in biotechnology. Different questions to be studied are the way the protein recognizes a particular ligand, how the ligand binds with the protein active site, and what conformational changes both molecules undergo during the ligand's entrance and exit. Such information allows us to predict the possibility of association between protein-ligand pairs, the strength of this association, or the protein activity level. Unfortunately, current experimental methods to obtain accurate (atomic-scale) information about protein-ligand interactions are extremely limited. Moreover, the large size of the search space to be explored and the long time-scales to be simulated are extremely challenging for the application of computational methods. This is especially true when full flexibility of the protein is taken into consideration.

Some software packages for predicting protein-ligand docking are available such as AutoDock [Goodsell 96], DOCK [Lang 09], FleX [Rarey 96], GOLD [Jones 97b] and ICM [Abagyan 94]. These packages use algorithms such as Monte Carlo, Molecular Dynamics, Genetic Algorithms [Goldberg 89], and fragment-based search [Hajduk 07] (for a survey of methods and software packages see [Sousa 06]). However, none of these software tools considers full flexibility of the protein. Moreover, these methods focus on finding the final binding conformation disregarding the ligand access/exit pathway, and without computing the conformational changes required for enabling such access/exit. An example of such protein-ligand accessibility problems is illustrated in Figure 1.8. Next, we survey works that use motion planning inspired methods for predicting binding sites and for computing access/exit ligand pathways.

Predicting Binding Sites

The algorithm introduced by Singh *et. al.* in [Singh 99] was tested on the following three protein-ligand complexes: *lactate dehydrogenase* with *oxamate*, *tyrosyl-transfer-*

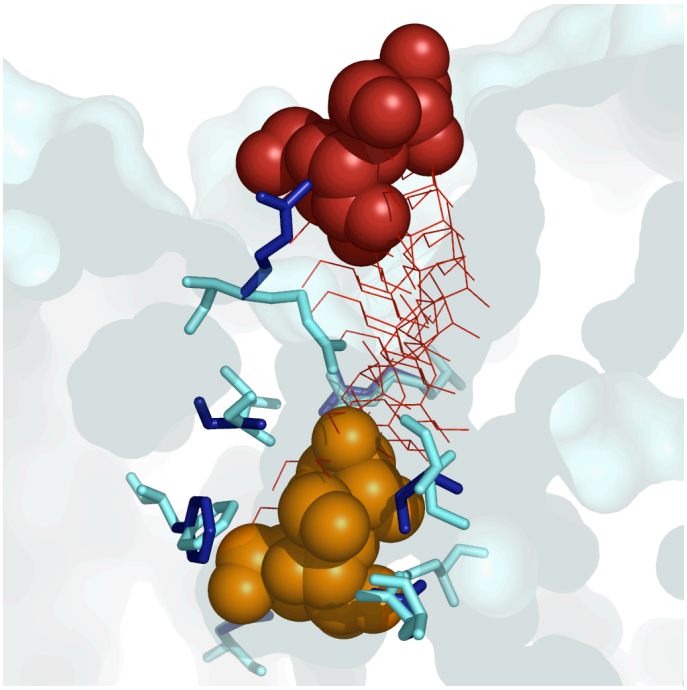

Figure 1.8: Illustration of protein-ligand accessibility problem. The figure shows a transversal cut of a protein with a ligand (represented with spheric atoms) occupying different locations: in the active site (orange) and on the surface (red). Some intermediate conformations of the ligand along the exit path are represented with red lines, and some side-chains that change their conformation during the ligand exit are represented with blue sticks.

RNA synthetase with *L-leucyl-hydroxylamine* and *streptavidin* with *biotin*. The algorithm was able to find the true binding site for the first two complexes successfully, but not for the third one. Such partial success corresponds to the overall performance of state-of-the-art methods.

More recently, Stochastic Roadmap Simulations have also been used in the study of protein-ligand interactions. In [Apaydin 02], SRS was applied to estimate the *escape time* for a ligand from different putative binding sites in a protein. Here, escape time is the expected amount of time for the ligand to escape from the "funnel of attraction" at the binding site [Apaydin 02]. Tests were performed on seven different protein-ligand complexes and results showed that, in five out of seven complexes, the escape time proved to be a good metric for distinguishing the catalytic site from the other putative binding sites. It is noteworthy to say that in both this work and in [Singh 99], only the ligand was assumed to be flexible and the protein was assumed to be rigid. This is possibly one of the reasons why these methods sometimes failed to predict correct binding sites.

Finding Access and Exit Pathways

The RRT-based method presented in [Cortés 05b] was applied to compute geometrically feasible paths of (R, S)-enantiomers to exit the active site of *Burkholderia cepacia lipase* (BCL). The flexibility of the ligand and of 17 side-chains in the catalytic pocket of BCL were considered. Energy profiles along the path were obtained by performing a rapid local minimization of intermediate conformations. Results showed a clear similarity between the computed paths and paths obtained using a pseudo-molecular dynamics approach. Remarkably, the combined RRT-minimization approach only required a few minutes to compute the paths, whereas pseudo-molecular dynamics took several days. Results also showed that the approach is suitable for pointing out protein residues that constrain the access of the ligand, which is highly valuable information for site directed mutagenesis. Further investigations about the influence of ligand access/exit on *Burkholderia cepacia* lipase enantioselectivity are presented in [Guieysse 08, Lafaquière 09]. These works show the ability of RRT-based methods to rapidly produce results that present fair qualitative agreement with experimental studies.

The extended ML-RRT method described in [Cortés 10b], able to deal with the protein backbone flexibility, was applied to compute the exit pathways of a bound substrate homolog (TDG) from *lactose permease* (LacY) and of *carazolol* from the active site of the β_2-*adrenergic receptor*. The considered molecular models involved several hundreds of degrees of freedom, and solution paths were obtained in several minutes. Results showed a remarkably good agreement with experimental data, as well as with results obtained with other, much more computationally expensive methods based on molecular dynamics.

1.4 Conclusion

We have surveyed the literature for methods based on robot motion planning algorithms to solve different problems in computational structural biology. The reviewed algorithms can be grouped based on the types of problems they have been applied to as shown in Table 1.1. We have also pointed out the main challenges and issues that need to be taken into account when extending motion planning methods for molecular simulations. A suitable representation for the molecule needs to be adopted, and an appropriate distance metric needs to be used for comparing molecular conformations. An efficient method for computing distances between atom pairs and for collision checking also needs to be considered, as well as a method for sampling conformations that satisfy structural constraints. Moreover, the ever-lasting problem of high dimensionality has to be faced, and an appropriate compromise should be made between the number of considered degrees of freedom and the amount of accuracy sought. Last but not least, energy needs to be made into account, and a choice has to be taken for the type of force field to be used.

Application Domain	Related Work
Loop Motions	**RLG-RRT** [Cortés 04, Cortés 05b, Barbe 11], **LoopTK** [Yao 08].
Domain Motions	**NMA-RRT** [Kirillova 08], **PathRover** [Enosh 08, Raveh 09], **PDST** [Haspel 10].
Protein Folding/Unfolding	**SRS** [Apaydin 02, Apaydin 03, Apaydin 04, Chiang 06, Chiang 07], **PRM-FP** [Song 02, Song 03, Amato 03, Thomas 05, Tapia 07, Thomas 07, Tapia 10], **MaxFlux-PRM** [Yang 07, Li 08]
RNA Folding	**PRM-FP** [Tang 05, Tang 08].
Protein Structure Prediction	**FeLTr** [Shehu 10].
Protein-Ligand Interactions	**PCR** [Singh 99, Apaydin 01], **SRS** [Apaydin 02], **ML-RRT** [Guieysse 08, Lafaquière 09, Cortés 10b].

Table 1.1: Motion planning inspired methods classified according to application domains.

Works reviewed in this chapter show that algorithms originating from robotics are promising complementary methods to more conventional techniques in computational structural biology. Their strength lies mainly in their efficiency in exploring highly complex spaces. Compared to classical methods such as MC, sampling-based motion planning algorithms require fewer iterations to find conformational transition pathways or to obtain a representative ensemble of conformational states. An additional advantage of motion planning inspired methods is that they do not require a force-field to drive the exploration, unlike MD simulations. Therefore, different types of data, including simple geometric models, can be used to constrain or to bias the search. The use of simple models leads to general and fast computational methods able to explore large regions of the conformational space. Results of such exploration can be further refined and analyzed subsequently using more accurate energy models.

Motion planning inspired methods for molecular simulations are still in their early stage. They require more improvements and validation on larger classes of systems. Further tests on real application problems, in tandem with experimental methods, will provide important feedback to improve the computational methods. Further work is also needed on the characterization of the results provided by these algorithms, using concepts of statistical physics.

As we have shown in this survey, the classes of structural biology problems to which motion planning inspired methods have been applied are still limited, being mainly focused around protein/RNA flexibility and protein-ligand interactions. Nevertheless, we believe that the potential of these methods is larger, and that other applications could be

investigated in the future. Examples of other interesting problems in structural biology are the prediction of protein-protein interactions and the conformational analysis of large molecular assemblies.

Chapter 2

A Mechanistic Model for Proteins

This chapter introduces a mechanistic modeling approach for proteins. This approach is based on the idea of decomposing the protein into fragments that can be dealt with as short kinematic chains. Such a decomposition leads to a multi-level representation that allows working with the protein in a coarse-grained manner, which expectedly leads to performance gains. At the same time the low level (full-atom) details of the protein are not lost and can be generated from the high-level representation whenever needed. This kind of modeling provides also a unified approach for implementing different already-available and new simulation methods, as will be seen in the next two chapters.

We begin this chapter with a quick overview of the structure of proteins. Next, we present the basics of modeling kinematic chains. Discussion then proceeds to the presentation of the proposed model.

2.1 The Structure of Proteins

Proteins are fundamental to all living organisms. They play essential roles in most biological processes that take place in the cell. They can take the form of enzymes that catalyze biochemical reactions and that regulate the metabolism process. They can also take the form of antibodies that bind to foreign substances to neutralize them. They also participate in biological functions like cell signaling, signal transduction and ligand transportation. Proteins can also have structural roles by helping the cell maintain its shape and size, and by producing mechanical forces as in muscle cells and sperm cells.

Molecules that are made of repeating structural units are called *polymers*. In this sense, proteins are a special kind of organic polymers, where the repeating structural unit is an *amino acid* residue. They are made of one or more chains that can have up to thousands of amino acid residues (short amino acid chains with less than 50 residues are often referred to as *peptides*). A protein usually folds into a stable three dimensional conformation that largely determines its functional role. Yet, they are not restricted

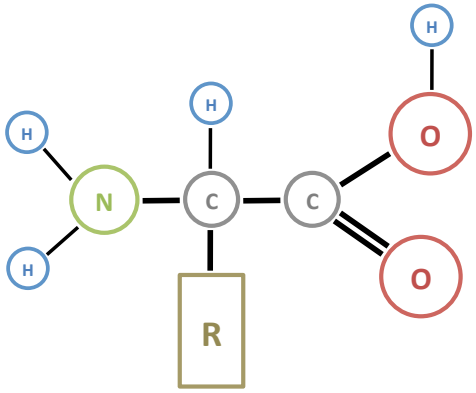

Figure 2.1: The chemical structure of an amino acid. The rectangle "R" resembles a side chain that differs from one amino acid to another.

by this native fold and usually undergo small to large conformational changes during biological processes.

Generally speaking, knowing the 3D structure of a protein helps in better understanding how it performs its function. Known structures of proteins can usually be found in the Protein Data Bank (PDB) [Berman 02], which is a free online repository that provides protein structures in the form of atom Cartesian coordinates. The great majority of protein structures available at the Protein Data Bank have been determined using X-Ray Crystallography [Woolfson 97], and most of the remaining structures have been determined using NMR Spectroscopy [Cavanagh 06]. Detailed statistics about the used experimental methods can be found at the PDB website[1].

The following paragraphs explain quickly notions related to protein structure that are needed for the discussion in this chapter and what follows. More detailed information can be found in text books on structural biology and protein structure prediction (the following are a few examples: [Banaszak 00, Schwede 08, Zaki 08, Sternberg 96]).

2.1.1 Amino Acids and the Primary Structutre

As shown in Figure 2.1, amino acids are chemically composed of a carbon atom (called C_α) that is connected to a carboxylic acid group (-COOH), an amine group (-NH2), a hydrogen atom and a side chain (R). Depending on the type of the side chain, amino acids show different physicochemical properties and are labeled with one of 20 names, which constitute the names of all the possible amino acid types naturally occurring in the proteins of living organisms.

Each pair of consecutive amino acids in a polypeptide chain is connected by a covalent bond (called the *peptide bond*) between the carbon atom in the carboxyl group of one amino acid and the nitrogen atom in the amine group of the adjacent amino acid. The

[1] http://www.rcsb.org/pdb/statistics/holdings.do

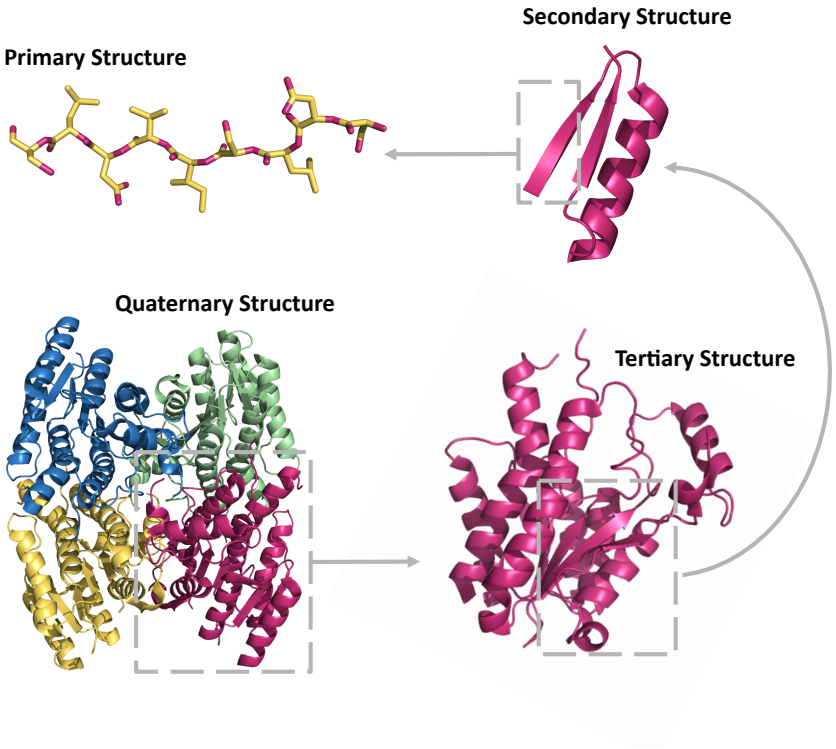

Figure 2.2: Relationship between the different levels of the protein structure hierarchy.

interaction between the carboxyl and amine groups causes an H_2O molecule to form and the carbon and nitrogen atoms to connect. Thus, the linear chain of amino acids has a carboxyl group (called the C-terminus) at one end and an amine group (called the N-terminus) at the other end, where in between, amino acid residues are connected with peptide bonds as explained. Atoms in the polypeptide chain, excluding atoms in side chains, are often referred to as the main chain or the protein backbone. The sequence of amino acids in a protein is usually referred to as the *primary structure*. This sequence is defined by the genetic code of a gene that is associated with the protein.

2.1.2 Higher-Level Structures

In addition to peptide bonds, hydrogen bonds also form between non-neighbor amino acids in the polypeptide chain. These bonds create reoccurring *secondary structure* local subunits that exhibit more structural stability than other parts of the polypeptide chain. There are two main types of secondary structure subunits: α-helices and β-sheets. Alpha-helices form due to interactions between close-by but non-adjacent amino acids in the same strand of the chain, which gives the strand the shape of a coil. On the other hand, β-sheets form due to interactions between amino acids in two parallel or anti-parallel strands in

the polypeptide chain. They are often represented in protein modeling software as arrows or long sheets. Examples of α-helices and β-sheets can be found in Figure 2.2. The remaining parts of the protein, which are relatively unstructured fragments, are called turns or loops.

The *tertiary structure* is the overall shape of a single polypeptide chain that shows how secondary structure subunits are connected and placed in reference to each other. This structure is largely determined by the primary structure. It forms due to long range and short range interactions that take place between different parts of the polypeptide chain, as well as due to interactions with the solvent. A single tertiary structure can be composed of several stable subunits called *domains* that are connected with more flexible links. These domains can have different functional roles during biological processes.

The final level in the protein structure hierarchy is the *quaternary structure*, which describes the overall arrangement of a protein complex, including all of its constituent polypeptide chains. These chains can be either repeated identical chains or different connected ones. Figure 2.2 illustrates the different types of protein structures and the relationship between them.

2.2 Proteins as Kinematic Chains

A kinematic chain is an assembly of rigid bodies, called *links*, that are connected by *joints*. This connection between rigid bodies creates motion constraints that need to be taken into account when modeling or dealing with the kinematic chain. The reason behind discussing kinematic chains in this section is that our proposed model, which will be introduced in the next section, considers proteins as kinematic chains. In the following, we quickly review basic notions in the modeling of kinematic chains and then show how they apply to the modeling of proteins.

2.2.1 Modeling Kinematic Chains

Rigid Bodies

In a three-dimensional (3D) Euclidean space $\mathbb{R}^3$, a single freely moving rigid body can be modeled by specifying the position and orientation of a reference frame attached to it. The position is conventionally specified using the cartesian coordinates, whereas there are several different methods for specifying the orientation. We restrict the discussion here to Euler angles [Taylor 05], which are among the most widely used methods for describing orientations. Hence, modeling the freely moving rigid body in $\mathbb{R}^3$ requires at least six independent parameters. We need three cartesian coordinates $\{x, y, z\}$ to describe the position, and three Euler angles $\{\alpha, \beta, \gamma\}$ to describe the orientation. These Euler angles are usually referred to as "*yaw*, *pitch* and *roll*".

To each rigid body in the space, we attach a Cartesian frame $F_{\mathcal{O}}$ that expresses the six parameters relative to a globally defined reference frame $F_{\mathcal{W}}$. The transformation of the coordinates from $F_{\mathcal{W}}$ to $F_{\mathcal{O}}$ can be expressed using the following homogeneous transformation matrix:

$$
{}^{\mathcal{W}}T_{\mathcal{O}} = \begin{pmatrix} \cos\beta\cos\alpha & \sin\gamma\sin\beta\cos\alpha - \cos\gamma\sin\alpha & \cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha & x \\ \cos\beta\sin\alpha & \sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & \cos\gamma\sin\beta\sin\alpha - \sin\gamma\cos\alpha & y \\ -\sin\beta & \sin\gamma\cos\beta & \cos\gamma\cos\beta & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.1}
$$

where the upper-left 3×3 sub-matrix defines the orientation of the rigid body and the upper-right 1×3 sub-matrix defines its translation. All points in a rigid body are assumed to have fixed coordinates relative to each other, which allows obtaining them directly from the attached frame.

Kinematic Chains

As mentioned before, a kinematic chain is composed of rigid links connected by joints. Depending on their types, these joints introduce different motion constraints to the rigid links. For example, the joint can be a simple *prismatic* joint that allows only a sliding motion along a single axis, or a *revolute* joint that allows a hinge motion relative to a single axis. The joint can also be a complex one that provides more than one degree of freedom, such as the *spherical* joint. However, complex joints with $n > 1$ degrees of freedom can generally be replaced by n consecutive simple joints of 1 degree of freedom.

To model the complete kinematic chain, a Cartesian frame is rigidly attached to each link and a transformation matrix is given between $F_{\mathcal{W}}$ and each of the attached frames. Although it is possible to perform all necessary operations using arbitrarily placed frames, following a systematic approach for placing these frames can simplify the performed operations. Therefore, we follow in this thesis the widely used *modified Denavit-Hartenberg* (mDH) convention [Craig 89], which is one of the most widely used conventions in robotics. In this convention all joints are assumed to be either prismatic or revolute, and z-axes are always chosen to be in the direction of the axes of the attached joints. Given two links with attached frames $F_{\mathcal{A}_i}$ and $F_{\mathcal{A}_{i-1}}$, the mDH convention also mandates the following two conditions:

- The axis x_i should be perpendicular to the axis z_{i-1}
- The axis x_i should intersect with the axis z_{i-1}

Exploiting these conditions, only four parameters are required for the modeling of the rigid links instead of six as mentioned before. Given the two frames $F_{\mathcal{A}_i}$ and $F_{\mathcal{A}_{i-1}}$,

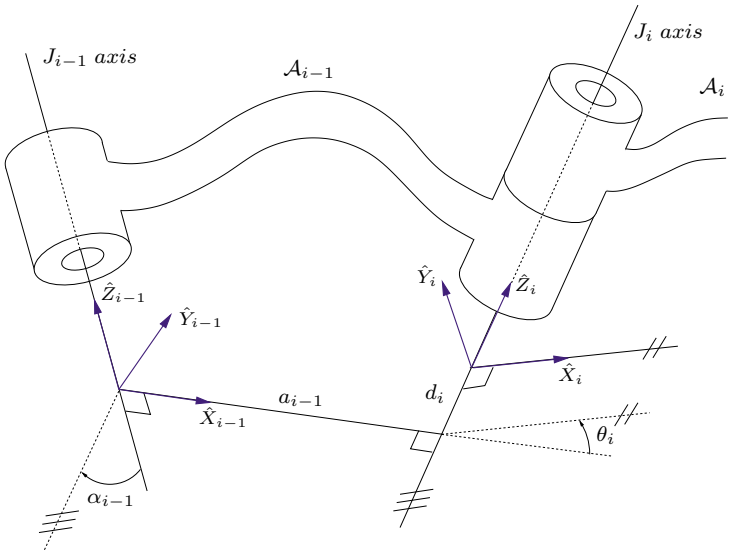

Figure 2.3: The mDH parameters defining the relative location of two links connected by a one-d.o.f. joint (following the convention in [Craig 89]).

these parameters are specified by the mDH convention as follows (see Figure 2.3 for an illustration):

- The *link length* (a_{i-1}) is the distance from z_{i-1} to z_i measured along x_{i-1}.
- The *link twist* (α_{i-1}) is the angle between z_{i-1} and z_i measured around x_{i-1}
- The *link offset* (d_i) is the distance from x_{i-1} to x_i measured along z_i.
- The *joint angle* (θ_i) is the angle between x_{i-1} and x_i measured about z_i.

Depending on the joint type, only one of these parameters is variable and all the rest are constant. If the joint is a revolute joint, then θ_i is the variable parameter, whereas if the joint is a prismatic joint, then the variable parameter is d_i.

Given the two links with attached frames $F_{\mathcal{A}_i}$ and $F_{\mathcal{A}_{i-1}}$, the location of $F_{\mathcal{A}_i}$ relative to $F_{\mathcal{A}_{i-1}}$ can now be given, according to the mDH convention, by the following homogeneous transformation matrix:

$$ {}^{i-1}T_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.2} $$

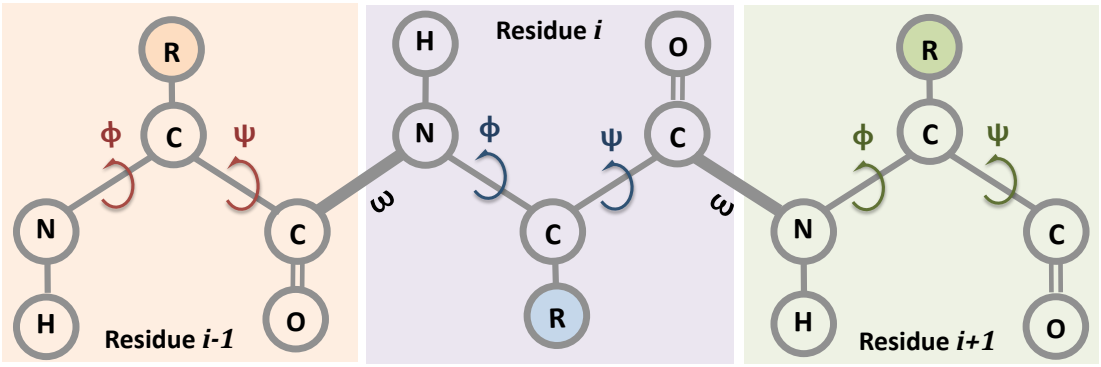

Figure 2.4: Dihedral angles in a polypeptide chain.

The location of a frame $F_{\mathcal{A}_j}$ relative to the base frame $F_{\mathcal{A}_0}$ can be computed from a sequence of local transformations as follows:

$$^{0}T_n = {}^{0}T_1 {}^{1}T_2 \dots {}^{n-1}T_n \tag{2.3}$$

where $^{0}T_1$ is a transformation that is equivalent to $^{\mathcal{W}}T_1$ since 0 is the index of the fixed base frame. More information about the modeling of kinematic chains can be found in text books about robotics such as [Xie 03, Angeles 07, Sciavicco 01].

2.2.2 Modeling Proteins

There is a direct correspondence between a polypeptide chain and a kinematic chain. Based on the internal coordinates representation and the rigid geometry assumption, both discussed in Section 1.1.2, a polypeptide chain is made of rigid atom-groups that are connected by bonds. Hence, bond torsions correspond to axes of revolute joints and atom-groups correspond to rigid links in the kinematic chain (a rigid body can be either an atom or a rigidly bonded group of atoms).

Figure 2.4 shows different dihedral angles in the backbone of proteins. These dihedral angles make the revolute joints in the kinematic model of the protein. In our work, we consider peptide bond angles, ω to be constant. This is because these dihedral angles are subject to very slight variations since peptide bonds are strong double bonds. Side chains can also be modeled in the same way. They are much shorter than the backbone and contain dihedral angles that are usually denoted as χ_1, χ_2, etc.

Using these notions and following the mDH convention, we can build a kinematic model for the protein as follows. A Cartesian frame $F_{\mathcal{A}_i}$ is rigidly attached to each rigid atom group $\mathcal{A}_i$ in the polypeptide chain. All frames are placed in a way that complies with the mDH conditions mentioned earlier. The relative location of the attached frames can then be expressed by the homogeneous transformation matrix defined in Equation

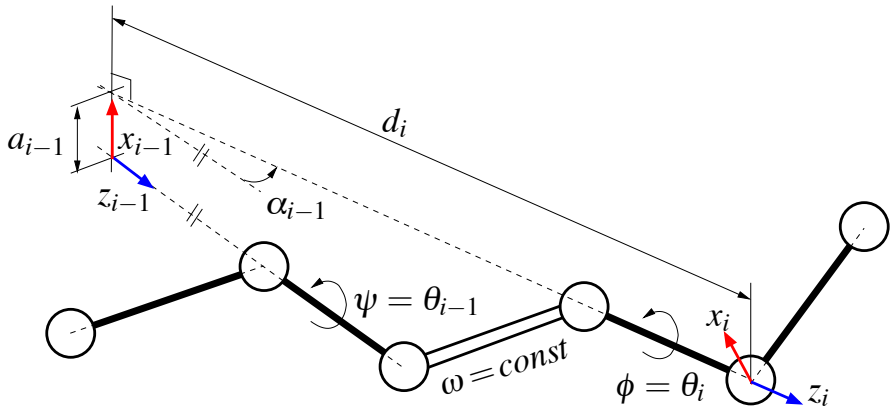

Figure 2.5: Kinematic model of the protein backbone around a peptide bond.

(2.2). Here the mDH parameters take the following meanings (See Section 1.1.2 for the definitions of the bond length, bond angle, and bond torsion):

- The *link length* (a_{i-1}) is the perpendicular distance between two successive bonds.
- The *link twist* (α_{i-1}) is the supplementary angle to the bond angle.
- The *link offset* (d_i) is the bond length.
- The *joint angle* (θ_i) is the bond torsion.

where θ_i is the only variable parameter since bond torsions correspond to revolute joint rotations as mentioned earlier. Figure 2.5 illustrates these parameters in the protein backbone.

2.3 Proposed Model

The protein modeling method we propose in this thesis is based on a multi-level modeling approach. It consists of a high-level decomposition of the protein into blocks of amino acids and a method for generating the low-level full-atom coordinates from this high-level decomposition. The following is an explanation of these two levels.

2.3.1 Decomposition Into Tripeptides

The main idea is to subdivide the polypeptide chain into fragments that contain exactly three amino acid residues each. We refer to these fragments henceforth as *tripeptides*. Depending on the number of residues in the polypeptide chain, which is often not divisible by three, this decomposition can yield at the end of the chain a fragment with less than three residues. This end fragment, regardless of its size, along with the first fragment in

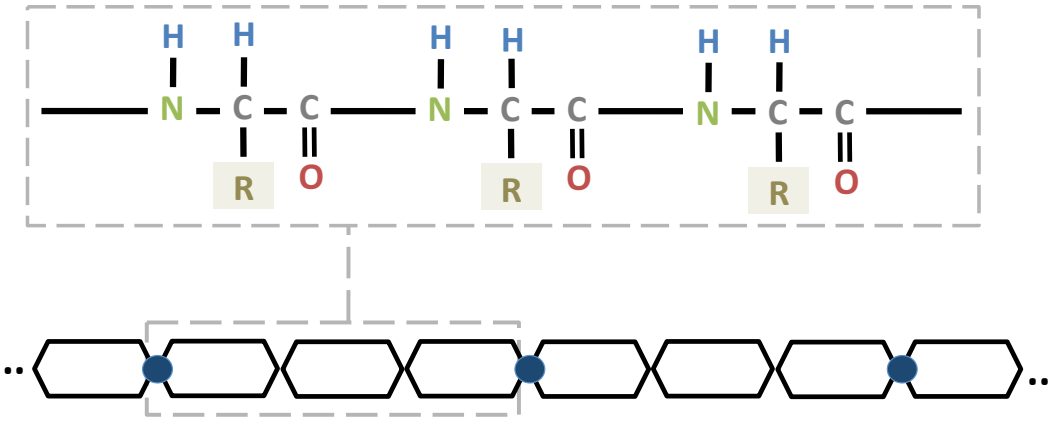

Figure 2.6: An illustration of a polypeptide chain subdivided into tripeptides. Blue circles represent particles and the highlighted rectangle shows the chemical composition of one tripeptide.

the chain are not considered as tripeptides in our model and require special treatment. These fragments of the chain differ from the other inner fragments (tripeptides) in that they are free at the N-terminal and C-terminal ends, which makes their movement less restricted than that of the tripeptides.

Each tripeptide in our model can be seen as a robotic manipulator with six revolute joints (i.e. with six degrees of freedom). This is because every tripeptide has three amino acid residues and every residue has two movable dihedral angles (ψ and ϕ) in its backbone. We attach a Cartesian frame to each atom group in the tripeptide as discussed in the previous section, however, we particularly label certain frames that are important for our model. These frames are the first and last frames in every tripeptide, which correspond to the base and end frames of the robotic manipulator. We refer to base frames henceforth as (oriented) *particles*. End frames at each tripeptide can be computed from the particle of the next tripeptide using constant transformations since tripeptides are connected by rigid peptide bonds as mentioned earlier. We refer to the model of the protein that includes only its particles as the *simplified particle-set model*.

Figure 2.6 shows part of a polypeptide chain that is subdivided into tripeptides, where the chemical composition of the tripeptide is shown in the highlighted rectangle and particles are depicted as blue circles. Figure 2.7 also shows an illustration of the proposed model applied on an SH3 domain (PDB ID: 1V1C). Figure 2.7.a shows the protein model with a ribbon representing the backbone embedded in the model of the protein surface. Figure 2.7.b represents the protein backbone trace with the frames corresponding to the particles. Figures 2.7.c and 2.7.d represent respectively the chemical and the mechanistic models of the backbone of a tripeptide.

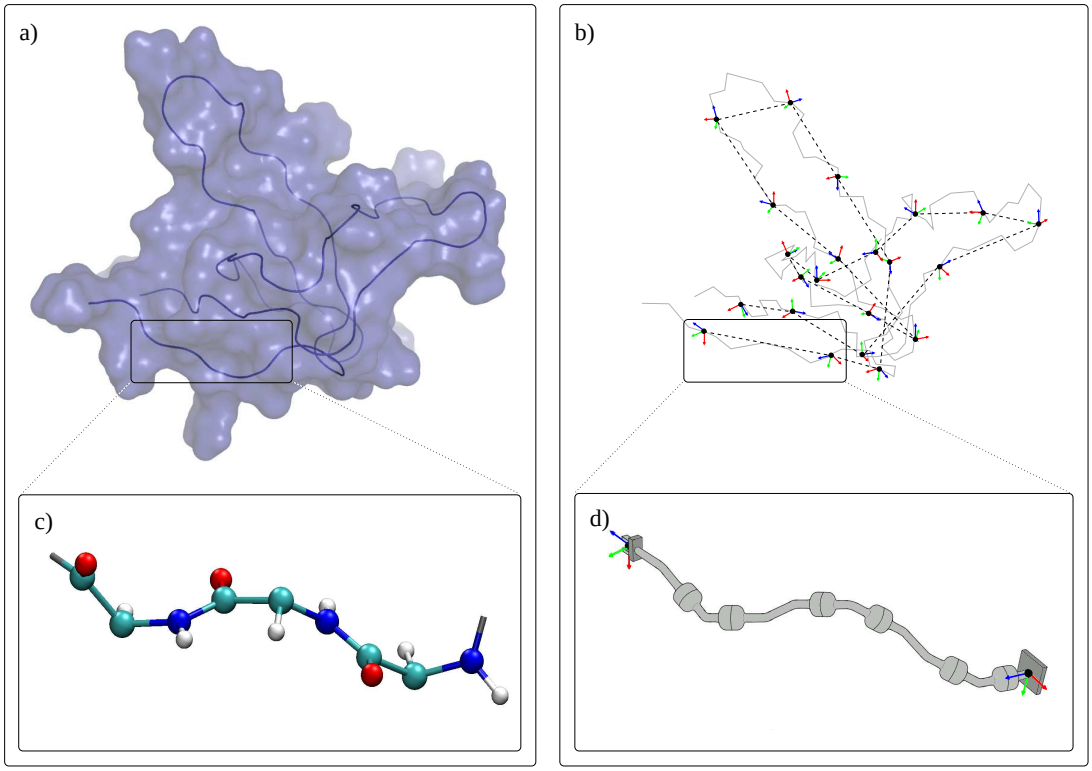

Figure 2.7: An illustration of the proposed approach. Tripeptides of three amino acid residues are treated as kinematic chains similar to robotic manipulators.

The main idea behind the proposed model is to enable sampling, deforming and generally treating the protein using only its simplified particle-set model rather than having to directly manipulate all of its atoms. Given a spatial configuration of the particle set, generating the corresponding values of dihedral angles at each tripeptide (and consequently the full atom model) can be done using *inverse kinematics* as will be discussed in the next section. The reason behind choosing to subdivide the protein into tripeptides with six dihedral angles is that the tripeptide is the shortest fragment with full mobility of the end-frame relatively to the base-frame. In other words, given the base frame, the end frame requires at least six dihedral angles in order to have the ability of adopting all the possible poses.

2.3.2 Solving Inverse Kinematics for a Tripeptide

The inverse kinematics (IK) problem for a kinematic chain is defined as "finding the values of the joint variables given the position and orientation of the end-effector relative to the base and the values of all of the geometric link parameters" [Siciliano 08]. In our case, it is the problem of finding the values of the ψ and ϕ angles in the backbone of a tripeptide given the pose of the particles.

As described in [Siciliano 08], there are two types of methods for solving inverse kinematics problems: closed form methods and numerical methods. Closed form methods are faster and can find all solutions that exist, however, they are not general, but robot dependent. These methods are restricted to systems with at most six degrees of freedom and whose geometries conform to certain conditions. Closed form methods use algebraic or geometric techniques in order to describe the problem in the form of a solvable system of equations. Conversely, numerical methods are not robot-dependent and can solve larger and more general systems. However, such methods are slower, can suffer from convergence issues and may not always be able to find all the existing solutions. Examples of such methods include [Canutescu 03, Zhao 94]

The method applied in this work for solving the IK problem for a general 6R serial kinematic chain is a closed form method that has been adapted from the solver proposed by Renaud [Renaud 00, Renaud 06]. This solver is based on algebraic elimination theory, and develops an ad-hoc resultant formulation inspired by the work of Lie and Liang [Lee 88b, Lee 88a]. Starting from a system of equations representing the IK problem (the formulation involves the product of homogeneous transformation matrices), the elimination procedure leads to an 8-by-8 quadratic polynomial matrix in one variable. The problem can then be treated as a generalized eigenvalue problem, as was previously proposed by Manocha and Canny [Manocha 94], for which efficient and robust solutions are available [Golub 96]. Our implementation applies the Schur factorization from LAPACK [Anderson 99]. Technical details on the applied IK solver are provided in the technical report of Renaud [Renaud 06].

This solver has been successfully applied in previous works on protein and polymer modeling [Cortés 04, Cortés 10a]. The advantage of this semi-analytical method with respect to numerical (optimization-based) methods, such as CCD [Canutescu 03], is that it provides the exact solution in a single iteration, not suffering from numerical convergence issues. The solver is very computationally efficient, requiring about 0.2 milliseconds on a single processor. Note however that our approach is not dependent on this solver, so that other IK methods (e.g. [Manocha 94, Coutsias 04]) could be applied.

There are often several possible solutions for a single base-end-frame pair, however, the number of possible solutions for kinematic chains with at most six degrees of freedom is known to be finite [Craig 89]. In fact, there are no more than sixteen unique solutions for such chains, provided that the joints are all revolute joints [Siciliano 08]. Depending on the application at hand, there are several possible strategies for choosing a solution out of the sixteen possible solutions of the IK problem. For example, a solution may be chosen at random if the application relies on randomness such as in Monte Carlo simulations. It is also possible to choose a solution corresponding to the best-energy conformation, especially in applications where maintaining a low energy profile is important or in appli-

cations whose goal is to find an energy minima. Another strategy is to choose the closest solution to the previous conformation of the tripeptide if the application prefers short moves over large jumps. In all cases, a filtering step may be required in order to ensure that solutions that cause steric clashes between different parts of the polypeptide chain are discarded.

Chapter 3

Enhancing the Monte Carlo Method

This chapter presents an example application for the use of the tripeptide model introduced in the previous chapter. It shows how this model can be used to facilitate the implementation of well-established Monte Carlo move classes as well as new ones. This flexibility of the model allows introducing higher level Monte Carlo sampling schemes that alternate between several implemented move classes, which enhances the overall performance of the method.

The first section of this chapter gives an overview of the Monte Carlo method and the following section discusses the implementation of new Monte Carlo move classes using the tripeptide model. The last section is dedicated to the discussion of Monte Carlo simulations that we have performed using the tripeptide model and the new move classes.

3.1 Overview of the Monte Carlo Method

The Monte Carlo (MC) method [Landau 05, Metropolis 53] is one of the most common computational techniques for studying molecules. It is mainly used for analyzing the energy landscape of the molecule and for computing thermodynamic properties such as average energy and heat capacity. Unlike Molecular Dynamics (MD) [Rapaport 07], which simulates deterministically the motion of atoms based on physical computations of the forces between them, MC explores the conformational space of the molecule through a random walk. This random walk favors low energy transitions and produces conformations that have a higher probability of being adopted by the molecule. Hence, MC is not capable of providing time-dependent quantities but is more efficient than MD in estimating average thermodynamic properties [Leach 01].

Starting at some initial conformation of the molecule, the Monte Carlo method iteratively performs a trial move by randomly perturbing the current conformation. The trial

move is then accepted or rejected based on a probability that takes into consideration its potential energy compared to that of the current conformation. This procedure produces conformations that form a *Markov chain*, since each trial move depends only on the current state and not on the other previous steps.

One of the most widely used MC acceptance probabilities is the Metropolis Criterion [Metropolis 53], which defines the probability of moving from state i to sate j as:

$$P_{ij} = \begin{cases} 1 & if\ E_j < E_i \\ e^{-\frac{E_j - E_i}{kT}} & if\ E_j \geq E_i \end{cases} \tag{3.1}$$

where E_j is the energy of the trial move, E_i is the energy of the current conformation, k is the Boltzmann constant and T is the temperature at which the simulation is performed. This probability function directly accepts moves that cause a decrease in potential energy and favors moves that cause a small increase in energy over those that cause high energy jumps.

A key issue to be addressed when performing simulations using the Monte Carlo method is the perturbation scheme applied (i.e. move class), especially in large molecular systems such as proteins. The used move class affects the efficiency of the exploration and the portion of the conformational space explored. Generally speaking, an effective move class should produce good coverage of the space with a good acceptance rate. A good acceptance rate saves the simulation from performing many useless energy computations.

One of the simplest and most frequently used type of moves are *pivot moves* [Lal 69]. These moves modify a single dihedral angle at a random position in the polypeptide chain, which causes all the atoms at one of the sides of this dihedral angle to be rotated as a rigid body accordingly. Such moves are numerically simple and generally effective. However, they can perform poorly in highly packed protein conformations, since small moves can cause large changes at the end of the chain, leading to a high rejection rate. Moreover, these types of moves cause atoms at the end of the polypeptide chain to move more frequently than middle ones.

To overcome the shortcomings of pivot moves, *local moves* have been introduced, which modify an arbitrary segment of the polypeptide chain while keeping all other parts of the chain intact. An example of such moves are the *Concerted Rotation* moves, which have been first proposed by [Gō 70] and then improved in [Dodd 93] and modified to satisfy detailed balance. These moves modify seven consecutive dihedral angles by rotating the first one (called the driver) and then adjusting the rest of the six dihedrals to accommodate this rotation without breaking bond constraints. A variant of this method has been introduced in [Leontidis 94] that generalizes it to more than seven dihedral angles.

Another example of local moves are CRRUBAR moves [Betancourt 05], whose name stands for *closed rigid-body rotations under bond-angle restraints*. These moves rotate

a window of an arbitrary number of consecutive residues around an axis between two backbone atoms. They promise a gain in performance over other move types, however, this comes at the expense of using variable bond angles, which contrasts the assumption followed in pivot and concerted rotation moves.

3.2 Devising Move Classes Using the Tripeptide Model

This section presents a unified approach for devising different move classes based on the tripeptide model introduced in the previous chapter. The common factor between the presented move classes is that they all rely on direct perturbation of the position and orientation of particles, and consequently, on the use of inverse kinematics to find conformations for tripeptides that are affected by the perturbation. As mentioned in Section 2.3.2, the use of inverse kinematics allows us to find a geometrically valid conformation for a tripeptide given the pose of the two particles attached to it.

There are several possible schemes for perturbing particles, depending on the number of particles involved, the perturbation method used, and the presence of a bias or of a motion correlation between the particles. We show in the following how to implement three simple and general-purpose move classes. Based on the tripeptide model, these classes, along with other conventional move classes like pivot and concerted rotations moves, can be easily combined using a higher-level sampling scheme that alternates between them, as will be explained at the end of this section.

One Particle Moves

The most straightforward move class using the tripeptide model is the perturbation of one particle (i.e. the perturbation of its pose). Such a perturbation requires adjusting the conformation of the two tripeptides whose end and base frames define the particle. This can be achieved by solving inverse kinematics for each of the two tripeptides. Hence, this move class introduces modifications to exactly twelve consecutive dihedral angles in the backbone of the protein. Figure 3.1 illustrates the idea.

This move class is expected to have a similar effect to other local, fixed-end move classes (e.g. [Gō 70, Dodd 93, Leontidis 94, Wu 99]). One advantage of this move class compared to other local and non-local move classes, is that a bias in Cartesian coordinates can be introduced easily to the perturbation of the particle depending on the application. This is especially interesting in applications where certain parts of the molecule are known to deform in a certain direction due to an interaction with another molecule for example.

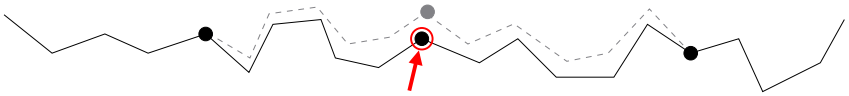

Figure 3.1: An illustration of the perturbation of one particle.

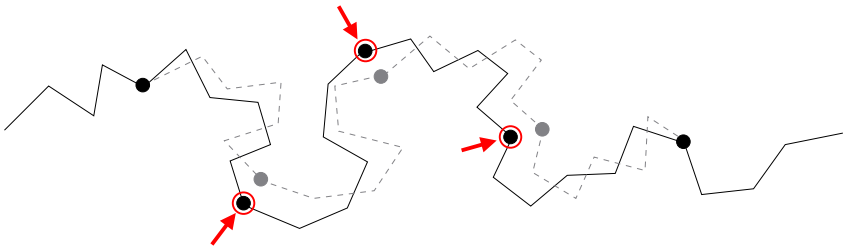

Figure 3.2: An illustration of the perturbation of three consecutive particles.

Flexible Fragment Moves

A simple extension to the *one particle* move class is to perturb a number of consecutive particles instead of only one. Note that perturbing n particles in random directions requires solving inverse kinematics $n+1$ times in order to adjust the conformation of all the tripeptides that are linked to the perturbed particles. An example of this move class with three particles is illustrated in Figure 3.2.

This move class has a similar effect to moves that are based on the cyclic coordinate descent method (CCD) [Canutescu 03], which are useful for perturbing flexible fragments of proteins. Generally speaking, these methods work by breaking the flexible protein fragment into two parts, where the dihedral angles in one part are perturbed and CCD is used to find a valid conformation for the second part in order to close the loop. The move class introduced here provides more flexibility in perturbing the fragment by direct manipulation of the particles and by introducing a bias for some or all the particles if desired.

Rigid-body Block Moves

Unlike the flexible fragment moves, which perturb n particles independently, this move class perturbs all the n consecutive particles together as a rigid body. In other words, all the n particles are translated and/or rotated around an arbitrary axis while preserving their positions and orientations relative to each other. Hence, the conformations of the tripeptides between these particles do not change. Nevertheless, the conformations of the tripeptide before the first particle and the tripeptide after the last particle need to

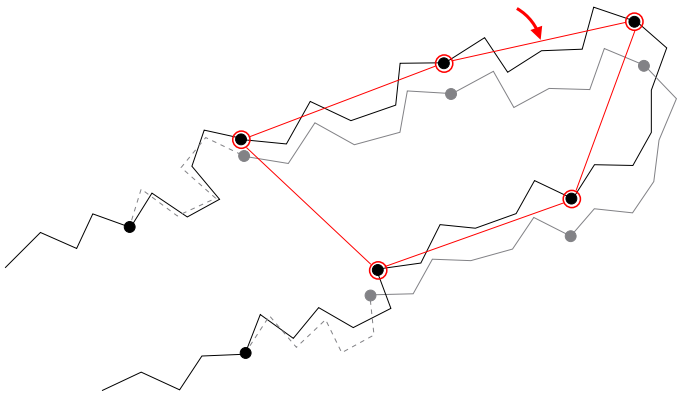

Figure 3.3: A rigid body rotation of a segment containing five particles around an axis defined by the two particles before and after the segment. This rotation simulates a hinge motion.

be adjusted using inverse kinematics. Figure 3.3 shows an example of this move class that resembles a hinge motion, where an axis of rotation is defined by two particles. Note that this move class is most similar to CRRUBAR moves [Betancourt 05], where a segment of the chain is rotated around an arbitrary axis defined by two atoms. Although the proposed method involves more complex algebraic operations than CRRUBAR, it presents the advantage that bond angles do not need to be perturbed and that it can be easily implemented using the tripeptide model.

Mixing Move Classes

One of the advantages of the proposed tripeptide model is that it provides a unified approach for implementing several move classes. This allows us to easily create a high-level sampling strategy that makes use of more than one move class. The rationale behind implementing such a high-level strategy is that using more than one move class introduces more variability to the sampled moves, which will expectedly lead to a better coverage of the conformational space. Performing different types of moves allows exploring a wider variety of paths and thus helps in overcoming energy barriers. This is clearly illustrated in the results presented in the next section.

There are many possible ways for combining move classes, which is something that is subject to investigation depending on the application at hand. One strategy, for example, is to randomly choose between the different available move classes using a uniform distribution. The strategy may also use probabilities that favor certain move classes over others. It can also equally mix sidechain and backbone moves, or use a probability that samples sidechains more frequently as they are known to be more flexible. Algorithm 3.1 shows the general steps of the mixing strategy.

Algorithm 3.1: MONTE_CARLO_WITH_MIXED_SAMPLING

```
input  : Initial conformation c_init, A set of n move classes M
output: A sequence of conformations C
begin
    C ← ADD(c_init);
    while not STOPCONDITION() do
        c_old ← LASTCONF(C);
        if ISSAMPLESIDECHAINS() then
            c_new ← SAMPLESIDECHAINS(c_old);
        else
            m ← SAMPLEMOVECLASSES(m);
            c_new ← SAMPLEBACKBONE(m, c_old);
        if METROPOLISTEST(c_old, c_new) then
            C ← ADD(c_new);
end
```

3.3 Experiments and Results

As a proof of concept, we show and discuss, in the following, results for several MC simulations implemented using the tripeptide-based model. These results are not meant to provide new insights into the proteins used in the simulations nor to prove the superiority of an MC move class over another. The aim of the experiments is to show that the tripeptide-based model provides flexibility in implementing new MC move classes, which can lead to a clear performance gain.

3.3.1 Experimental Setup

We have performed four sets of MC simulations using two different proteins (two sets for each protein). Each set of simulations consists of five independent MC runs performed using five different move classes. Details of these simulations are described in the following.

Proteins Used

We have chosen for our tests two small proteins that are topologically different. The first protein is the *SH3 domain* of obscurin. This protein is composed of 68 amino-acid residues and has five β-sheets connected by relatively long loops. It can be found in the Protein Data Bank under the ID: 1V1C. The second protein is an intrinsically disordered protein called *Sic1 protein*. It is composed of 77 residues and it lacks any type of secondary structures (except for one nigligibally small α-helix). The model of the protein was generated using the Flexible-Meccano method [Bernadu 05] for sampling a statistically probable backbone conformation, and SCWRL4 [Krivov 09] for the side

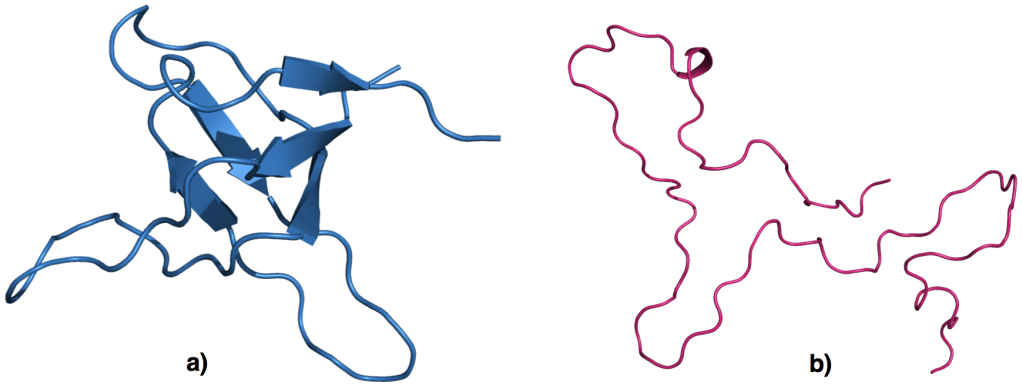

Figure 3.4: Proteins used in the simulations. The SH3 domain is shown in sub-figure (a) and the Sic1 protein is shown in sub-figure (b)

chains. An illustration of the two proteins can be found in Figure 3.4

Move Classes

We have implemented the following five move classes:

- *OneTorsion*: This is the simplest type of moves. It involves rotating a randomly chosen dihedral angle from the backbone of the protein and then propagating the motion to the end of the chain as shown in Figure 3.5. This move corresponds to the previously mentioned *pivot moves*.

- *ConRot*: This move class is based on the *Concerted Rotations* of Dodd *et al.* [Dodd 93]. We have implemented this move class using the tripeptide model as follows. A dihedral angle is chosen at random from the backbone of the protein and is randomly perturbed. Assume that this dihedral angle lies at tripeptide T_i. The motion caused by the perturbation is then propagated to the end of the tripeptide T_i. Next, the conformation of the tripeptide T_{i+1} is adjusted by solving IK between the new pose of the end of T_i and the original pose of the end of T_{i+1}. This move class is depicted in Figure 3.6

- *OneParticle*: This move class corresponds to the one particle move described in the previous section. A randomly chosen particle is perturbed and the conformation of each of the two adjacent tripeptides is found using IK.

- *Hinge*: This move class corresponds to the rigid-body block moves described in the previous section. First, a random starting particle p_i and a random segment length l are chosen. The segment length l is chosen such that it is always larger than 3 and less than a certain predefined constant N. Next, particles between p_i and p_{i+l}

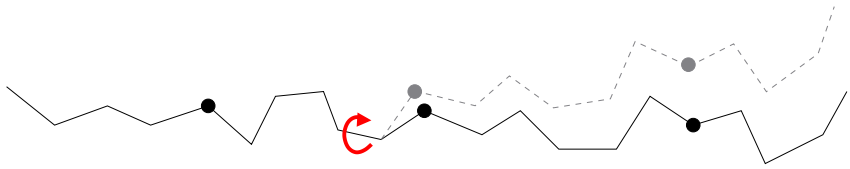

Figure 3.5: An illustration of a *OneTorsion* move.

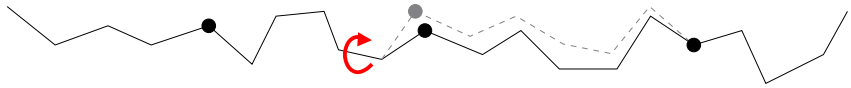

Figure 3.6: An illustration of a *ConRot* move.

are all rotated (with a random value) as a rigid body simulating a hinge movement. The rotation is performed around the axis defined by the two particles p_i and p_{i+l}. Conformations of the tripeptide before p_i and the tripeptide after p_{i+l} are then adjusted using IK.

- *Mixed*: This is simply a mix of the four previous move classes. At each iteration of the MC method, one of the four move classes is randomly chosen and applied as shown in Algorithm 3.1. Function SampleMoveClasses chooses uniformly between the four different classes and function IsSampleSideChain gives equal weights to performing backbone and sidechain moves.

Parameteriztion

For each of the two proteins, we have performed two sets of simulations using different step sizes. Table 3.3.1 shows the different step sizes used. These step sizes indicate maximum perturbations applied to the original values at each move trial. In other words, if the current value is x and the step size is s, then a random value is chosen in the interval $[x - s, x + s]$. Step sizes have been chosen such that they produce comparable atom displacements in a chain fragment, and at the same time, comparable acceptance rates.

Each simulation set includes five independent MC runs using the five previously mentioned move classes. Before running these simulations, the two proteins were equilibrated by running an MC simulation using the *oneParticle* move class and *step-size I* (see Table 3.3.1). This simulation was run until 10^5 trial moves were accepted. Starting from

		Bond Torsions	Particle Translations	Particle Rotations
SH3	Step I	0.03 rad.	0.1 Å	0.01 rad.
	Step II	0.09 rad.	0.3 Å	0.03 rad.
Sic1	Step I	0.09 rad.	0.3 Å	0.03 rad.
	Step II	0.15 rad.	0.5 Å	0.05 rad.

Table 3.1: Sets of perturbation step sizes.

the equilibrated conformations, each MC simulation was run until 10^6 trial moves were accepted. Each iteration in all the performed simulations includes the following:

- Randomly choosing between either performing a backbone move or a side-chain move.
- If a side-chain move is to be performed, then all the dihedrals at a randomly chosen side-chain are perturbed.
- If a backbone move is to be performed, then the designated move class of the simulation is applied.
- After performing any move (side-chain or backbone move), the conformation of the protein is checked for collisions. Non-bonded atoms are considered to be in collision if the distance between them is less than 70% of the van der Waals equilibrium distance [Bondi 64].
- Trial moves that fail to find IK solutions or that produce conformations with self-collisions are directly rejected.
- Energy is evaluated for collision-free conformations and the Metropolis Criterion is applied to accept or reject them.

All hinge moves have been performed with a segment length that contains between 4 and 10 particles. On the other hand, all simulations have been performed at a temperature of 300K and all energy evaluations have been performed using the OPLS-AA force field [Jorgensen 96] together with an implicit representation of the solvent using the Generalized Born approximation.

3.3.2 Results

We discuss in the following the results of the performed simulations from two perspectives:

- **Efficiency of the exploration**: Table 4.5 shows computational results that describe the efficiency of the exploration for each simulation. It shows the number of energy evaluations performed, the CPU time required to complete the simulation and the acceptance rate, which equals to the number of conformations (10^6) divided by the number of trials. Note that simulations were run on a single AMD Opteron 148 processor at 2.6 GHz. Generally speaking, an efficient move class has a high acceptance rate (for a comparable step-size), which implies performing less energy evaluations. Such an efficiency strongly affects the CPU time required for the method to sample a number of states.

- **Quality of the exploration**: Quality of the explored set of conformations can be inferred from the structural distances traveled (structural dissimilarity) and the energies of the set of samples provided by the MC method. Figures 3.7, 3.8, 3.9 and 3.10 show two types of plots. The first type depicts how the average distance of sampled conformations from the initial conformation evolves over time, where the distance is the root mean square deviation (RMSD) of the dihedral angles. The second type of plots shows how the average energy of sampled conformations evolves over time. Good coverage implies a higher average of distances and a lower average of energies.

Efficiency of the Exploration

Looking at the computational results in Table 4.5, it is clear that the *OneTorsion* move class is outperformed by all the other move classes in all the simulations in terms of CPU time. The reason for this is two fold. First, although perturbations performed by the *OneTorsion* move class are small, they require all the coordinates of the atoms following the perturbed dihedral angle to be recomputed. This requires more time than updating only two tripeptides in the case of *ConRot* and *OneParticle*, even though IK computations are involved. The second reason, which is more significant, is clear from the poor acceptance rate of *OneTorsion* compared to the other methods, which led to a higher number of energy evaluations. This is mainly due to the aforementioned propagation, which makes the move susceptible to producing large displacements far-away from the perturbed dihedral angle. Unlike local *oneParticle* moves, such global moves produce high energy fluctuations that are more difficult to accept. The performance of *OneTorsion* could have been even worse relative to the other move classes if a more intelligent energy computation method had been used. Currently, energy is completely reevaluated after each move, where it could have been reevaluated by considering only parts that have

changed. This reduces the computation time in local moves such as *oneParticle* and *conRot* compared to global moves like *oneTorsion*.

On the other hand, *ConRot* and *OneParticle* exchanged turns across simulations of the two proteins. *ConRot* was the best in terms of CPU time in the SH3 simulations, whereas in the Sic1 simulations *OneParticle* was the best. Nevertheless, a clear advantage of the *OneParticle* move class can be outlined. Looking at the results, *OneParticle* simulations performed the least number of energy evaluations in all the simulation sets except the first one (SH3 with Step Size I). This is even true for cases when the acceptance rate of *OneParticle* was not the best, such as in SH3 Step Size II and Sic1 Step Size I. Note that the acceptance rate depends on finding IK solutions, collision-free conformations as well as low energy transitions. This feature is especially important for large proteins, where the cost of energy evaluations is higher and the use of an efficient exploration method is imperative.

Finally, simulations with the *Mixed* sampling strategy have achieved intermediate CPU time results compared to the other move classes. This was expected since this sampling strategy invokes all the other four move classes, which makes its performance affected by both the best and worst methods. The *Hinge* move class also produced staggering results, which means that it requires more care in setting its parameters depending on the topology of the molecule.

Quality of the Exploration

Looking at Figures 3.7 through 3.10, it is clear that the relationship between the computational results and the quality of the exploration is not straight forward. For example, although *OneTorsion* performed worst than all the other move classes in all the simulations as mentioned before, this did not necessarily translate to a lagging exploration in all the simulations. Similarly for *ConRot* and *OneParticle*, their aforementioned leading positions did not grant them equivalent positions in the distance and energy plots. In fact, these three move classes did not show any repeating pattern over the four sets of simulations.

The main reason behind these observations is that high quality exploration generally depends on the ability of the method to produce diverse fluctuations that allow visiting different areas in the conformational space. A high acceptance rate may lead to producing more quickly a larger number of conformations, however, these conformations may all be from the same local region. Structural fluctuations need to be large enough to move out of the vicinity of the starting conformation, and small enough at the same time to avoid introducing large energy fluctuations that slow down the exploration. Hence, move classes can perform very differently depending on the step sizes and the topology of the molecule, which mandates careful parameter setting.

This discussion leads us to the explanation of the persistently leading performance of the *Mixed* sampling strategy, which is evident in all the distance and energy plots. Simulations performed with the *Mixed* strategy show a profile of average distance from the initial conformation that is higher than all the other move classes. This means that this strategy is able, on average, to visit conformations farther than any of the other four move classes. At the same time, simulations also show that the *Mixed* strategy is able to maintain an average energy profile that is lower than all the other move classes. This leading performance is a direct consequence of the diversity of the structural fluctuations achieved by this sampling strategy, since it alternates between four different move classes. This makes it relatively more capable of exploring the conformational space regardless of the step size used. Importantly, this good exploration does not come at the expense of high performance requirements, as the *Mixed* strategy maintains a fair computational performance compared to the other move classes as discussed previously.

3.4 Conclusion

The main power of the tripeptide-based protein model is that it provides a unified approach that enables implementing many of the widely used move classes as well as for devising new move classes, as seen in this chapter. Moreover, it simplifies implementing complex move classes, such as the *Mixed* sampling strategy, which offers a clear enhancement of performance in exploration over other move classes, maintaining at the same time average CPU time requirements. Using this *Mixed* strategy, simulations explore relatively better at a relatively fine speed, regardless of the step size used or what the topology of the protein is. This is, of course, true as far as the presented initial simulations show.

More experiments are needed to better validate these results and to test other variations of the mixed strategy. For example, it may be interesting to perform tests that can lead to knowing the optimal set of move classes. A move class may only be reducing the acceptance rate or increasing the computation time without adding real value to the quality of the exploration, which is something that needs further experiments to understand. It is also interesting to test different probabilities for choosing between the move classes and to try giving more weight to sampling sidechains over sampling the backbone.

Other possible experiments include testing the *flexible fragment* move class in different variants. For example, particles in the fragment can be perturbed all with the same step size, or conversely, with step sizes that are larger for inner particles and smaller for particles that are closer to the two ends of the fragment. The effect of replacing the *oneParticle* move class in the mixed strategy with the flexible fragment move class is also worth exploring.

	Move Class	Acceptance Rate	Energy Evaluations.	CPU Time
SH3 - Step I	*ConRot*	0.73	1.3×10^6	39 h.
	Hinge	0.73	1.3×10^6	41 h.
	Mixed	0.62	1.4×10^6	42 h.
	OneParticle	0.61	1.6×10^6	49 h.
	OneTorsion	0.58	1.7×10^6	57 h.
SH3 - Step II	*ConRot*	0.48	1.7×10^6	51 h.
	OneParticle	0.45	1.6×10^6	53 h.
	Mixed	0.42	1.9×10^6	57 h.
	Hinge	0.47	2.0×10^6	64 h.
	OneTorsion	0.40	2.3×10^6	68 h.
Sic1 - Step I	*OneParticle*	0.46	1.5×10^6	40 h.
	Hinge	0.52	1.7×10^6	45 h.
	ConRot	0.46	1.7×10^6	46 h.
	Mixed	0.43	1.8×10^6	53 h.
	OneTorsion	0.44	2.2×10^6	57 h.
Sic1 - Step II	*OneParticle*	0.43	1.6×10^6	42 h.
	ConRot	0.35	2.1×10^6	56 h.
	Hinge	0.39	2.2×10^6	58 h.
	Mixed	0.32	2.4×10^6	63 h.
	OneTorsion	0.32	2.9×10^6	74 h.

Table 3.2: Computational performance of the four simulation sets.

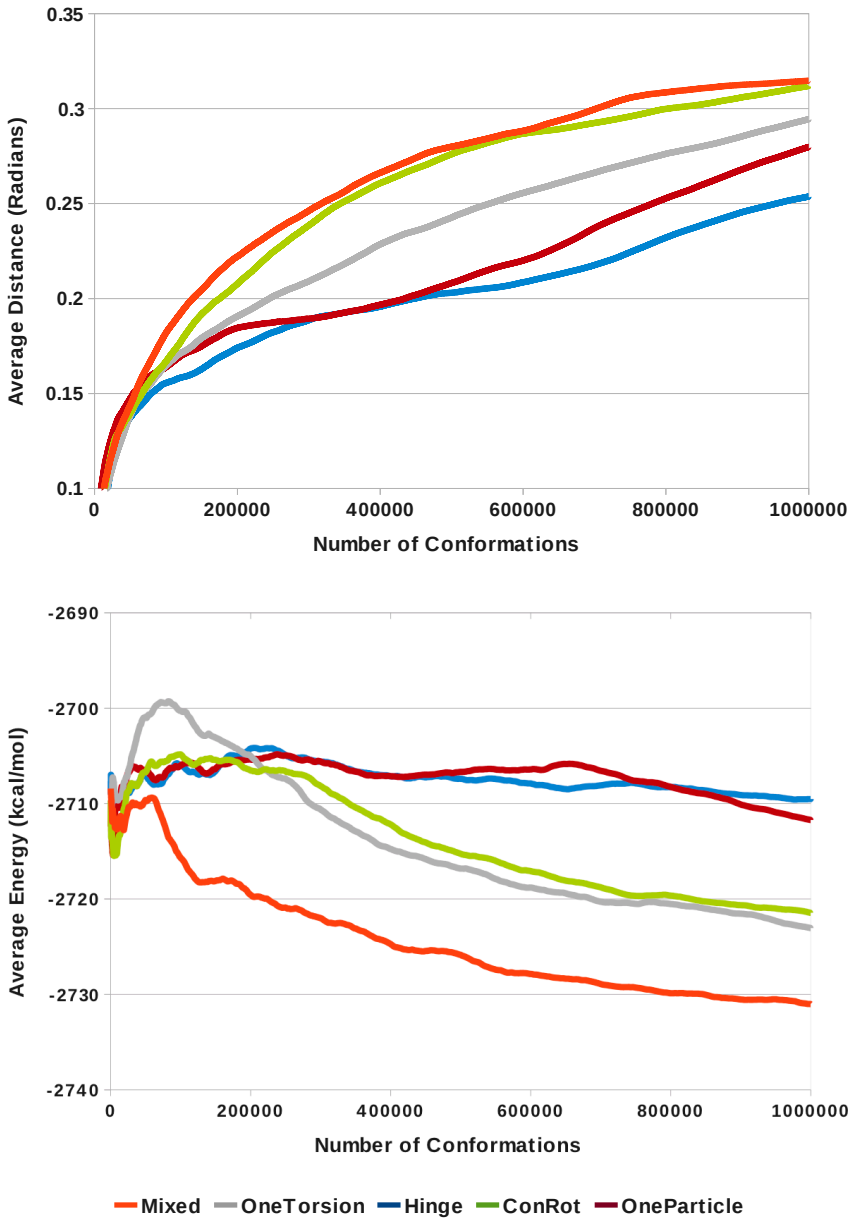

Figure 3.7: Evolution of the average distance and average energy over time in the simulations performed with the SH3 domain and step size I.

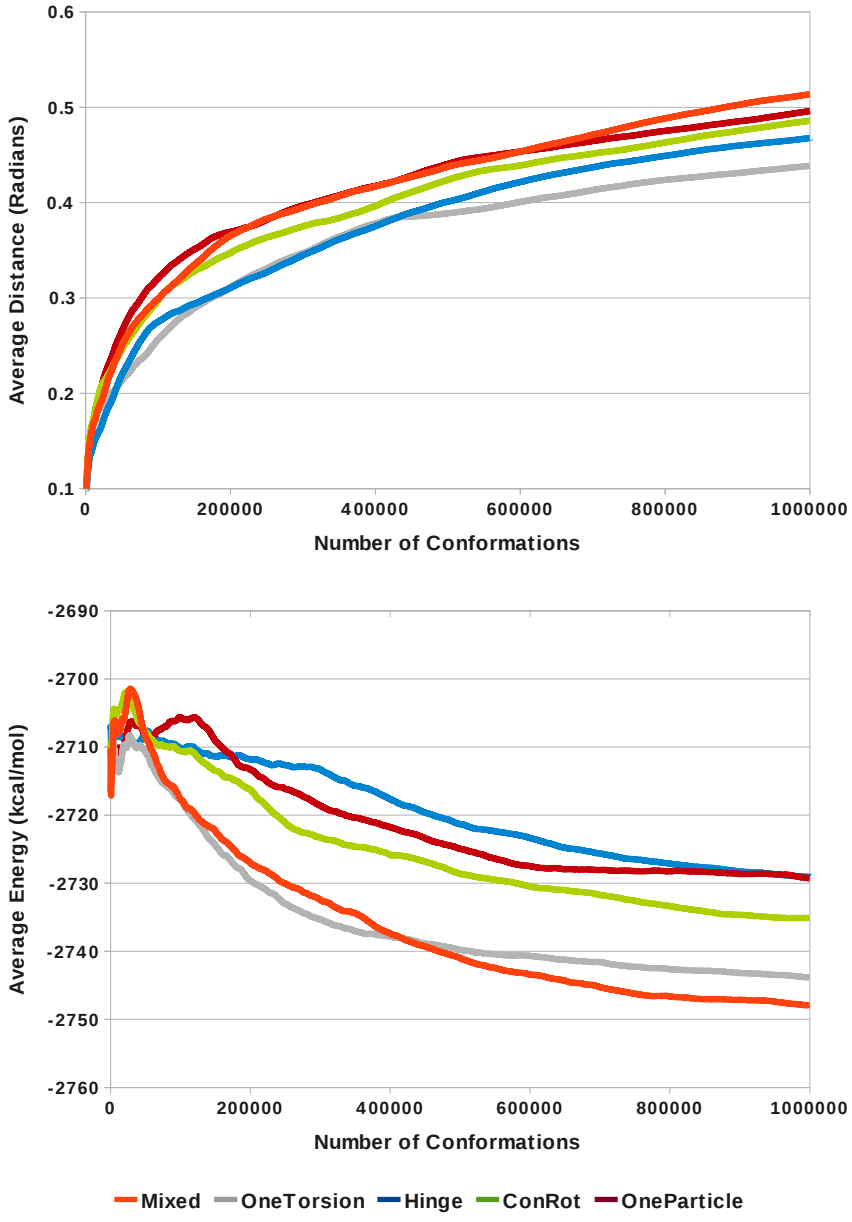

Figure 3.8: Evolution of the average distance and average energy over time in the simulations performed with the SH3 domain and step size II.

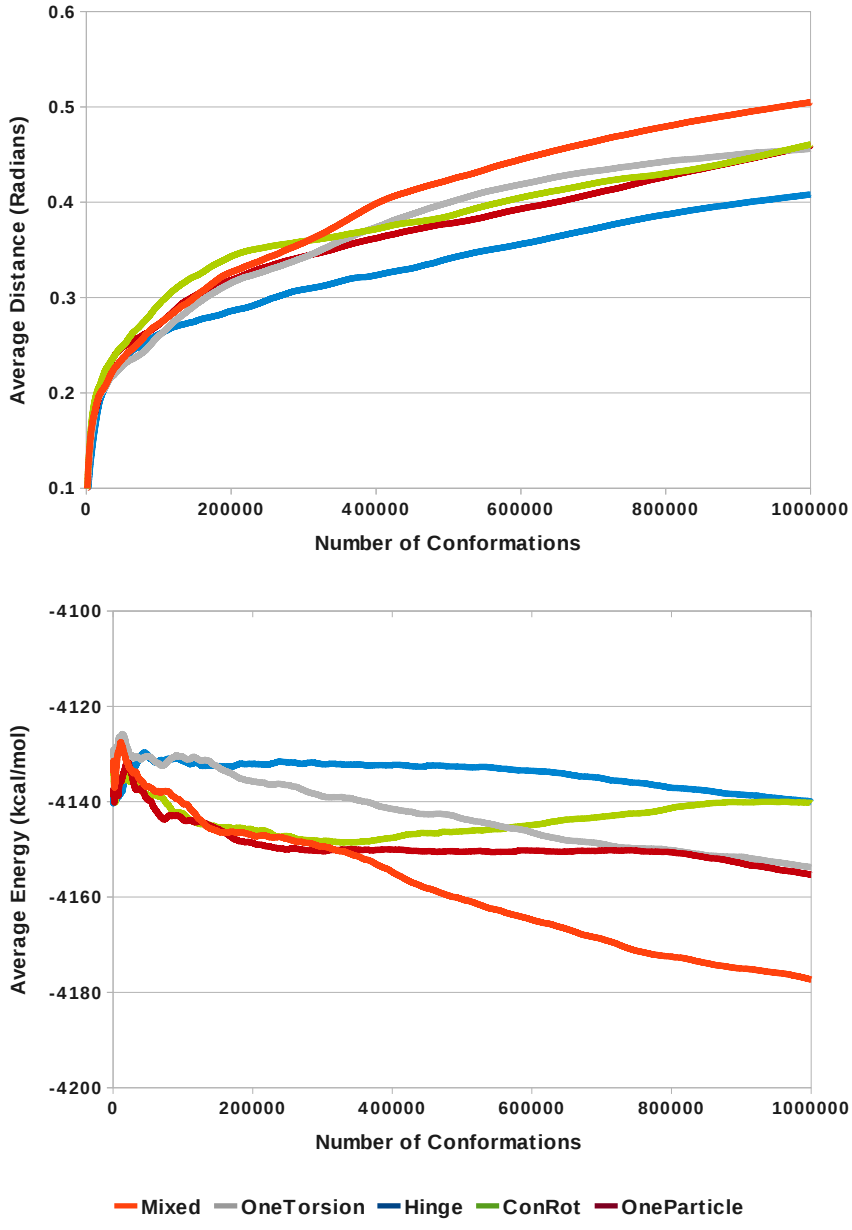

Figure 3.9: Evolution of the average distance and average energy over time in the simulations performed with the Sic1 protein and step size I.

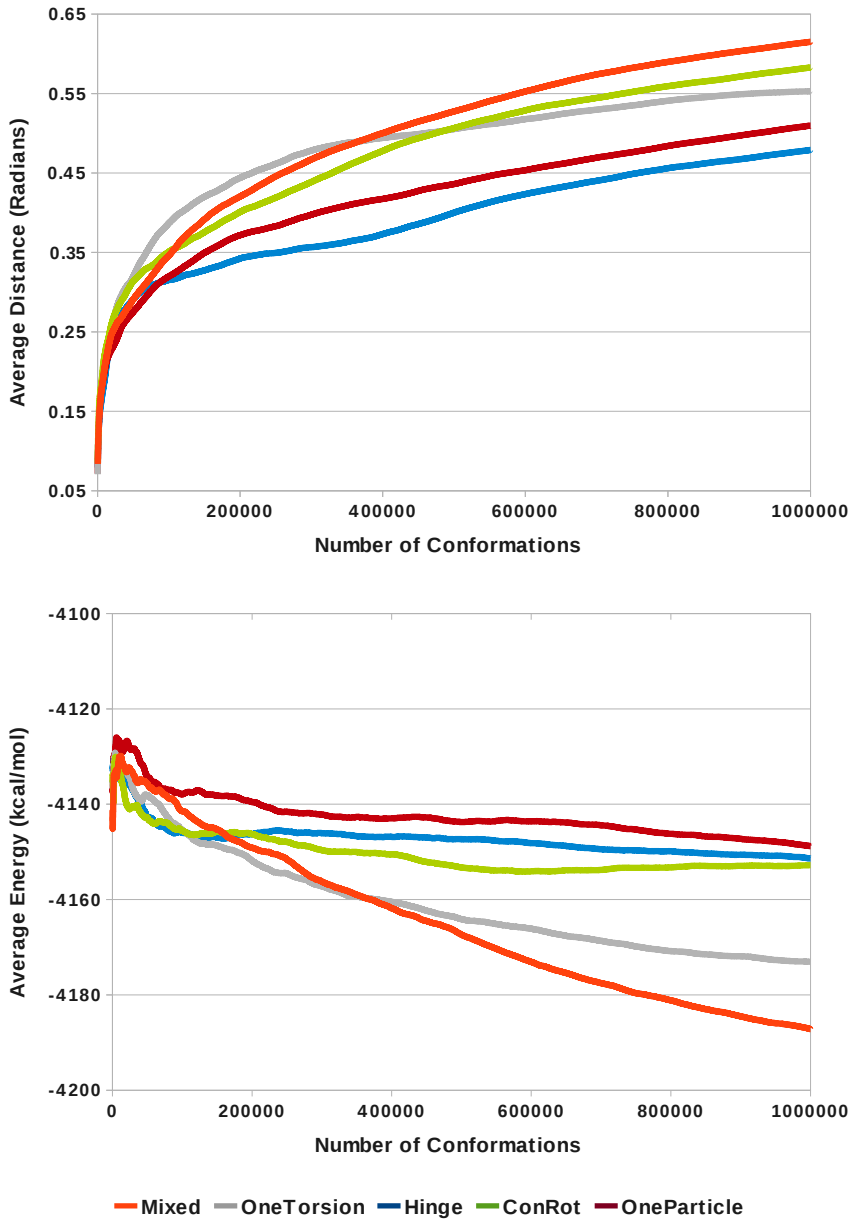

Figure 3.10: Evolution of the average distance and average energy over time in the simulations performed with the Sic1 protein and step size II.

Chapter 4

Exploring Conformational Transitions

4.1 Overview

This chapter introduces a new method for exploring the conformational space of proteins. The method is based on the tripeptide protein representation (discussed in Chapter 2) and applies a combination of the RRT algorithm and Normal Mode Analysis (NMA) [Cui 06]. This method is particularly useful for analyzing protein conformational transitions, especially those involving domain motions.

As mentioned in Section 1.3.1, studying conformational transitions in proteins is important for understanding their biological functions, since such motions are generally related to the capacity of the protein to interact with other molecules. However, capturing this type of dynamic information at the atomic scale is difficult using experimental methods. Therefore, computational methods like Molecular Dynamics and Monte Carlo are most commonly used. Nevertheless, these methods also suffer from efficiency problems when applied to compute large-amplitude conformational changes.

In this context, we propose a computational method that extends the methods introduced in [Cortés 05b, Kirillova 08]. These methods use an RRT to speedup the exploration of the conformational space, and thus, enable the simulation of large-amplitude protein motions with few computational resources. The method introduced in [Kirillova 08], goes even one step further and makes use of normal mode analysis to bias the search of the RRT towards energy-favorable regions, which allows studying problems with even higher dimensions. This idea of biasing the RRT using normal modes is rooted in works such as [Brooks 85, Hinsen 98, Tama 01, Alexandrov 05], which show the ability of normal modes to predict the direction of collective conformational changes (like domain motions) in macromolecules. However, since normal modes provide local predictions and not full conformational trajectories, iterative methods have been intro-

duced that perform short displacements and recompute the normal modes at each step [Mouawad 96, Miyashita 03, Jeong 06]. Such methods require a large number of iterations to compute large conformational transitions, which is something that can be avoided using RRTs as has been shown in [Kirillova 08].

The method proposed here also uses normal modes to bias the search of the RRT. However, the main difference with [Kirillova 08] is that our method is based on the simplified particle-set model. Such an apparently minor change has nonetheless important outcomes. Using this model the number of normal modes per protein is reduced at least by a factor of three, which greatly reduces the time required to compute them. Another advantage of using the tripeptide model is that it provides an accurate method for moving between the coarse-grained particle-based representation and the all-atom model. In [Kirillova 08], the sampling step of the RRT is performed in Cartesian coordinates (as it relies on the normal modes), whereas all the other steps are performed in internal coordinates in order to ensure producing conformations with valid bond angles and bond lengths. Therefore, a conversion step is repeatedly performed to move back and forth between the cartesian and internal coordinates. The approximation implied by the change of representation make the performance of the algorithm greatly dependent on the topology of the protein rather than on its size, which limits the use of the algorithm to certain types of proteins. In this chapter, we show how our algorithm overcomes this problem providing an efficient performance that linearly scales with the number of residues.

The main idea of this chapter is to show how the tripeptide-based model, the RRT algorithm and normal mode analysis can create an effective tool for studying conformational transitions, when combined together. Both, the tripeptide model and RRT, have been discussed separately in Chapters 1 and 2. Therefore, discussion in Section 4.2 will concentrate more on normal mode analysis and on how the three methods can be combined together. In Section 4.3, we present experiments performed with 10 different proteins of sizes between 214 and 994 residues, and provide a discussion of the achieved results.

4.2 Method

4.2.1 Elastic Network Models and Normal Mode Analysis

Every molecule has a set of natural vibration modes that depends on the structure of the molecule. Each mode corresponds to a pattern of motions, in which all atoms of the molecule move with the same frequency and phase, i.e. all passing through the equilibrium and maximum points at the same time. These modes are called *normal modes* and can be calculated by diagonalizing the Hessian matrix of the potential energy. It has been shown that low frequency normal modes correspond to collective atomic motions (or domain motions), whereas high frequency normal modes correspond to local

fluctuations [Atilgan 01, Gō 90, Hinsen 98].

The approach we adopt for computing normal modes in our method is a simplified one that is based on considering the molecule as an elastic network [Tirion 96]. The Elastic Network Model (ENM) represents the molecule as a set of nodes connected by springs. All the protein atoms can be considered as nodes in this model, however, a more coarse grained representation is usually applied that considers nodes to be only the C_α atoms [Tama 01]. Moreover, nodes are connected by virtual springs only if the distance between them is less than a user-defined cut-off distance D_{cut}. The potential energy function of such an elastic network takes the following form:

$$E = \sum_{d_{ij}^0 < D_{cut}} \frac{C}{2}(d_{ij} - d_{ij}^0)^2 \tag{4.1}$$

where d_{ij} is the distance between node i and node j, d_{ij}^0 is the distance between the two nodes at the equilibrium state and C is the elastic constant.

This type of simplified elastic networks has been used in many works and for very different applications [Kim 02, Tama 04, Cavasotto 05a, Jeong 06]. Here, we investigate going further in the simplification of the elastic network model. Instead of using C_α atoms, we build the ENM using the *simplified particle-set representation* (see Section 2.3.1), thus considering one node per tripeptide. As shown in [Tama 01], using a simplified ENM does not necessarily lead to a loss of accuracy in the prediction of motion directions, however, it certainly leads to more performance efficiency. This issue is discussed in more details in Section 4.3.1. Figure 4.1 shows a protein in the form of an elastic network that is built from the particles of the tripeptide model.

Normal modes in our method are computed as follows. First the *Hessian Matrix* of the elastic network is constructed from the particle positions. The Hessian matrix is the second partial derivative of the potential energy E, where each element in the matrix is a 3×3 matrix that corresponds to the interaction between two particles. Hence, the size of the hessian matrix is $3N \times 3N$, where N is the number of particles. Each 3×3 element can be computed as follows [Atilgan 01, Eyal 06]:

$$H_{ij} = -\frac{C}{d_{ij}^2} \begin{bmatrix} (x_j - x_i)(x_j - x_i) & (x_j - x_i)(y_j - y_i) & (x_j - x_i)(z_j - z_i) \\ (y_j - y_i)(x_j - x_i) & (y_j - y_i)(y_j - y_i) & (y_j - y_i)(z_j - z_i) \\ (z_j - z_i)(x_j - x_i) & (z_j - z_i)(y_j - y_i) & (z_j - z_i)(z_j - z_i) \end{bmatrix} \tag{4.2}$$

$$H_{ii} = -\sum_{j|j \neq i} H_{ij} \tag{4.3}$$

where, i and j correspond to particle indices. If the distance between particles i and j

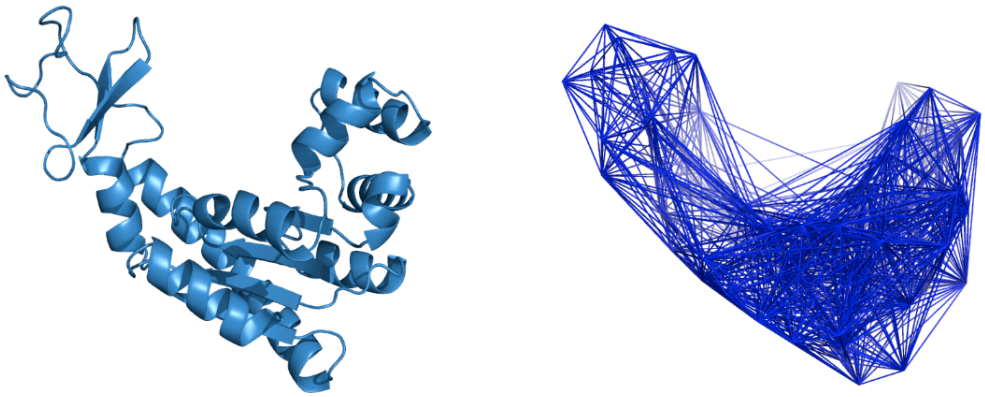

Figure 4.1: The ADK protein (PDB ID: 4ake) represented as an elastic network, where nodes are particles in the tripeptide model.

is more than the cut-off distance, then the whole 3×3 matrix is replaced by zeros. The Hesian matrix is then diagonalized to compute the eigenvalues and eigenvectors. Each eigenvalue and eigenvector pair corresponds to one normal mode, where the eigenvalue defines the mode frequency and the eigenvector defines displacements for each particle in the Cartesian space. Note that using particles instead of C_α atoms reduces the matrix size by a factor of 3, which greatly reduces computation time.

4.2.2 Overall Algorithm

The particular problem addressed in this chapter can be phrased as follows. Given a starting protein conformation q_{init} and an end conformation q_{goal}, the algorithm should produce a sequence of conformations $q_1, q_2, ..., q_{n-1}$, which the protein can possibly adopt as it moves from q_{init} to q_{goal}.

The proposed method works by iteratively creating short consecutive RRTs. Each iteration consists of: computing the normal modes from an initial conformation q_{root}, using these normal modes to bias a short RRT exploration, and then choosing a new q_{root} for the new iteration (q_{root} for the first iteration is q_{init}). Each RRT explores until it moves a predefined distance to the target q_{goal}. Once the RRT stops, the closest node in the tree q_{close} to q_{goal} is identified, and the path between q_{root} and q_{close} is extracted and saved. All nodes on this path are guaranteed to have a collision-free backbone (which implies having acceptable energy values), since conformations are accepted only if their backbone atoms are collision-free. In order to rearrange side-chains, a quick minimization step is performed on q_{close}, which is then used as the new root of the RRT in the next iteration and the base for the new computation of the normal modes.

The algorithm keeps iterating until a predefined distance d_{target} from q_{goal} is reached. The resulting trajectory is then defined by the minimized conformations q_{close} at each

Algorithm 4.1: COMPUTE_PATHWAY

```
input  : Initial conformation q_init, final conformation q_goal and minimum distance
         to target d_target
output: The transition pathway p
begin
    q_root ← q_init;
    while RMSD(q_root, q_goal) > d_target do
        m ← COMPUTE_NORMALMODES(q_root);
        t ← BUILD_RRT(m, q_root, q_goal);
        q_close ← CLOSESTTOTARGET(t, q_goal);
        q_root ← MINIMIZE(q_close);
        p ← CONCATENATE(p, q_root);
end
```

Algorithm 4.2: BUILD_RRT

```
input  : Initial conformation q_root, final conformation q_goal
output: The tree t
begin
    t ← INITTREE(q_root);
    while not STOPCONDITION(t, q_goal) do
        q_rand ← SAMPLE(t);
        q_near ← BESTNEIGHBOR(t, q_rand);
        q_new ← EXPANDTREE(q_near, q_rand);
        if ISVALID(q_new, t) then
            ADDNEWNODE(t, q_new);
            ADDNEWEDGE(q_near, q_new);
end
```

iteration. If a finer grained trajectory is required, then the extracted paths at each iteration can be used, which may require further minimization. The steps of the algorithm are summarized in Algorithm 4.1.

4.2.3 Particle-Based RRT

Each individual RRT in the sequence of executed RRTs in the proposed algorithm performs the same steps as the standard RRT steps described in Algorithm 4.2. However, the distinctive trait of these RRTs is that they sample the coordinate space of the normal modes instead of the space of the degrees of freedom of the protein. Another important difference is that these RRTs work on the simplified particle-set representation rather than directly on the atoms.

As described in Algorithm 4.2 and in Section 1.1.1, the first step at each iteration of the RRT algorithm is to generate a random conformation q_{rand}. This conformation acts

as a determinant of the direction towards which the tree is extended. Next, the tree is searched for a conformation q_{near}, which is the closest conformation in the tree to q_{rand}. A new conformation, q_{new}, is then generated by moving a predefined distance from q_{near} towards q_{rand}, and the new conformation is added to the tree if it does not violate any geometric constraints. The following are the details of how each of q_{rand}, q_{near} and q_{new} are generated in our method.

Sampling Random Conformations

The idea is to generate a random sample q_{rand} that allows the RRT to explore the conformational space using information given by the normal modes. To achieve this, the following steps are performed:

- A sequence of n random weights are sampled in the range of [-1, 1], where n is the number of particles multiplied by 3, which equals to the number of normal modes.
- Each weight w_i corresponds to a normal mode nm_i and is multiplied by an amplification factor f that is the same for all the normal modes. This factor is used to push the sampled conformation away from q_{root}.
- An array of $n/3$ particle positions is created by computing their positions from a linear combination of all the modes and their weights. More precisely, the x-coordinate of a particle i is computed as follows:

$$p_{ix}^{new} = p_{ix}^{old} + \sum_{j<n}^{j=0} w_j * f * nm_j \tag{4.4}$$

  where p_{ix}^{old} is the x-coordinate of the particle i in q_{root}

The resulting array of particles acts as q_{rand} in the following steps of the algorithm. This is because it contains all the necessary information for finding q_{near} and generating q_{new}. Hence, q_{rand}, in our case, is not an all-atom conformation, but a list of particle positions. These positions have been created by moving the original particles found at q_{root} in the directions given by a normal combination of normal modes with randomly sampled weights.

Finding Nearest Neighbors

In order to find q_{near}, the tree is searched for the conformation that is closest to q_{rand}. The distance is computed between every conformation in the tree and q_{rand}, where the computed distance is the root mean squared deviation (RMSD) between the particle

positions. An additional bias is also used in our algorithm to pull the exploration towards the goal conformation. This bias is introduced to the computed distance as follows:

$$distance(q_i, q_{rand}) = \frac{RMSD(q_i, q_{goal})}{RMSD(q_{init}, q_{goal})} RMSD(q_i, q_{rand}) \tag{4.5}$$

In other words, the node that is both closest to q_{rand} and to q_{goal} is favored over other nodes. The node with the minimum distance is chosen as q_{near} which is then extended towards q_{rand} in order to generate q_{new}. In this work, we have implemented a simple brute-force algorithm to find q_{near}. However, more sophisticated nearest neighbor search algorithms based on space partitioning techniques (e.g. [Atramentov 02]) could be used to speed up the process and reduce the number of performed distance measures.

Generating New Conformations

In order to generate q_{new}, all particle positions in q_{near} are linearly interpolated towards q_{rand} with a predefined distance k. Given these interpolated particle positions, the full atom model of q_{new} can be generated using inverse kinematics. We apply an iterative process that solves inverse kinematics for every tripeptide t_i using the two interpolated particles p_i and p_{i+1}. If no IK solution is found for a tripeptide or if the solution found is not collision-free, we slightly perturb the attached particles and try again. A small perturbation is also applied to the particles' orientations, since the cause of the problem can be due to restraints caused by the current orientations of the particles. This process is repeated until a collision-free IK solution is found or a maximum number of trials has been reached.

If this process fails to find a collision-free IK soltuion for any tripeptide, failure is reported and the RRT algorithm goes back to the random sampling step. After generating IK solutions for all the tripeptides, the only remaining parts of the protein to be addressed are the two end-fragments attached to the first and last tripeptides. The pose of these fragments is adjusted such that they are in accordance with the new poses of the first and last particles respectively. The pose is also adjusted such that changes in the first and last tripeptides are propagated to these fragments. A random perturbation can also be applied to the two end fragments depending on the application.

The generated conformation q_{new} is guaranteed to satisfy hard geometric constraints. As mentioned before, every generated tripeptide conformation is checked for self collisions for collisions with other parts of the protein. However, in order to reduce the rejection rate, sidechains are excluded from the collision checking (only C_β atoms are considered). This is because sidechains are known to be very flexible, and resolving their collisions is easier than resolving collisions in the backbone. Hence, any sidechain collision is assumed to be resolved during the minimization step at the end of each short RRT execution, as mentioned in Section 4.2.2.

4.3 Experiments and Results

This section discusses experiments that we have performed to validate the performance of the proposed method. First, we begin by addressing the question raised in Section 4.2.1 about the effect of using a coarse grained elastic network model that is built using the tripeptide-based model. Next, we present experiments that show the good performance of the proposed method in exploring the conformational space to find conformational transitions in proteins.

4.3.1 Validating the Elastic Network

Previous works such as [Tama 01, Hinsen 98, Hinsen 99] have shown that simple ENMs built using C_α atoms perform as well as ENMs built using the all-atom model. This has been shown to be true as far as the study of dynamic properties in proteins is concerned. In the following, we compare the ability of ENMs built using the *simplified particle-set* model to the ability of ENMs built using the C_α atoms to predict motion directions during molecular transitions. For this, we use the notion of *overlap* as proposed in [Marques 95, Tama 01].

The overlap I_j between a normal mode j and an experimentally observed conformational change between two conformations (open and closed) q^o and q^c is defined as a measure of similarity between the conformational change and the direction given by the normal mode j [Tama 01]. It can be computed as follows:

$$I_j = \frac{\left| \sum^{3N} a_{ij} \Delta q_i \right|}{\left[\sum^{3N} a_{ij}^2 \sum^{3N} \Delta q_i^2 \right]^{1/2}} \tag{4.6}$$

where $\Delta q_i = q_i^o - q_i^c$, is the difference between the i^{th} atomic coordinates of the protein in conformations q^o and q^c, a_{ij} corresponds to the i^{th} coordinate of the normal mode j and N is the number of C_α atoms. In our case, the cartesian coordinates of particles i in the tripeptide model are used in Δq_i instead of C_α atoms, and N corresponds to the number of particles. A value of 1 for the overlap means that the direction given by the normal mode matches exactly the conformational change, whereas a value that is around 0.2 or less means that the normal mode is unable to give any meaningful prediction.

We have measured overlap values for the seven proteins shown in Table 4.1, which are proteins that have been used also in [Tama 01] for the validation of the C_α ENM. All the simulations in [Tama 01] have been performed using a cutoff distance of 8 Å as suggested in [Bahar 97]. A good cutoff distance should create an elastic network that correctly captures the topology of the protein. However, it can be intuitively inferred

that the same cutoff distance may not be the optimal choice in our case, since distances between particles of the tripeptide model are not the same as the distances between C_α atoms. Hence, we have measured and compared overlap values for the seven proteins with cutoff distances between 8 and 34 Å to find the optimal one.

Figure 4.2 shows the average overlap value achieved for each cutoff distance over the seven proteins. The overlap value considered for each protein is the best one found among the overlap values of all the normal modes. As can be clearly seen in the figure, the highest averages are for cutoffs 15, 16 and 17. This is expected since tripeptides have three C_α atoms each, and they usually adopt conformations that are not fully extended. This means that the optimal cutoff distance is expected to be less than three times the optimal cutoff used in C_α elastic networks.

In Table 4.2, we show overlap values using a cutoff distance of 16 Å and compare them to the overlap value presented in [Tama 01] for the C_α ENM. It is clear that both ENMs give comparable overlap values, which means that our simplified ENM is also able to capture the topology information necessary for producing normal modes that correctly predict motion directions. Table 4.3 shows that the overlap values can even be better when a different (best) cutoff distance is used for each protein separately. The presented values have been measured for both the case of moving from the open conformation towards the closed conformation and *vice versa*.

Protein	Residues	PDB ID_{open}	PDB ID_{closed}
Che Y Protein	128	3chy	1chn
LAO binding Protein	238	2lao	1laf
Triglyceride Lipase	256	3tgl	4tgl
Thymidulate Synthase	264	3tms	2tsc
Maltodextrine Binding Protein	370	1omp	1anf
Enolase	436	3enl	7enl
Diphtheria Toxin	523	1ddt	1mdt

Table 4.1: Proteins used in the *overlap* experiments.

Protein	C_α Overlap		Particles Overlap	
	Open	Close	Open	Close
Che Y Protein	0.32	0.34	0.52	0.34
LAO binding Protein	0.84	0.40	0.53	0.52
Triglyceride Lipase	0.30	0.17	0.26	0.35
Thymidulate Synthase	0.56	0.40	0.49	0.29
Maltodextrine Binding Protein	0.86	0.77	0.90	0.84
Enolase	0.33	0.30	0.40	0.30
Diphtheria Toxin	0.58	0.37	0.48	0.30

Table 4.2: Comparison between overlap values for ENMs built using the simplified particle-set model and ENMs built using C_α atoms as presented in [Tama 01]. The used cutoff distances are 16 and 8 for the two ENM types respectively. Columns labeled "Open" are for the case of moving from the open to the closed conformation and columns "Closed" are for the opposite case.

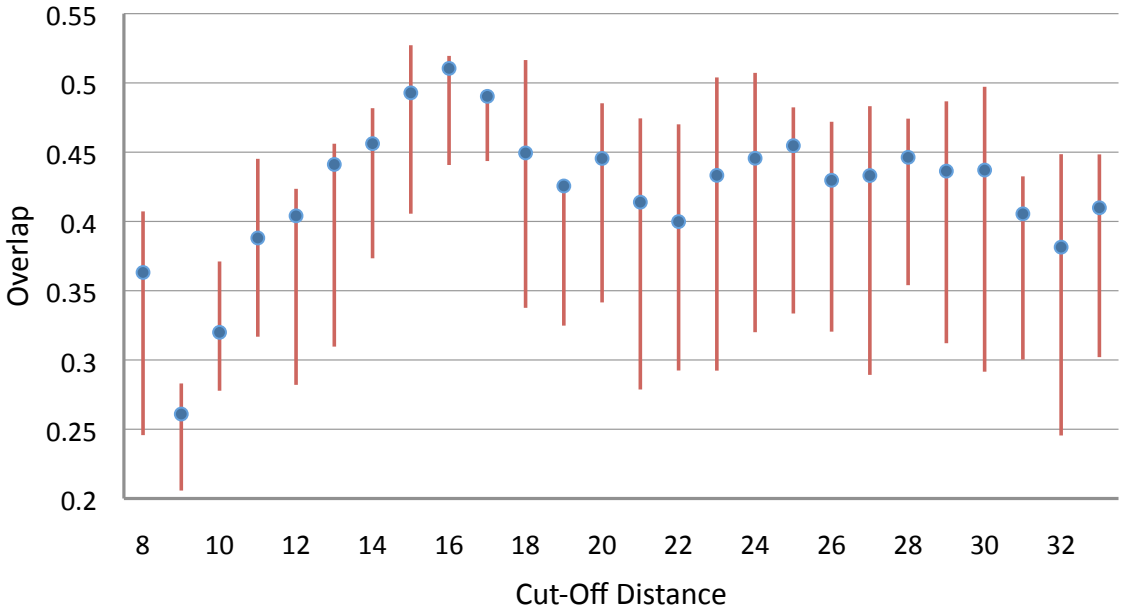

Figure 4.2: Average overlap over the seven proteins of Table 4.1. Each red line starts at the 25^{th} percentile of all the overlap values and ends at the 75^{th} percentile, where the blue circle marks the average overlap value.

Protein	Overlap_{best}		Overlap_{16}	
	Open	Close	Open	Close
Che Y Protein	0.76	0.64	0.52	0.34
LAO binding Protein	0.55	0.63	0.53	0.52
Triglyceride Lipase	0.45	0.42	0.26	0.35
Thymidulate Synthase	0.51	0.37	0.49	0.29
Maltodextrine Binding Protein	0.90	0.84	0.90	0.84
Enolase	0.45	0.45	0.40	0.30
Diphtheria Toxin	0.48	0.36	0.48	0.30

Table 4.3: Overlap_{best} is the best overlap value achieved using any cutoff distance between 8 and 34, whereas Overlap_{16} is measured using a cutoff distance of 16.

4.3.2 Finding Conformational Transitions

Experimental Setup

We have applied the proposed method to compute conformational transition pathways for the 10 proteins shown in Table 4.4. For each protein, there are at least two experimentally identified conformations, where the difference between these conformations involves large amplitude domain motions. The size of these motions varies depending on the protein, where the smallest motion is 2.75 Å C_α-RMSD in NS3 and the largest is 10.96 Å C_α-RMSD in DDT. All the studied domain motions are shown in Figures 4.7 through 4.16. We have also chosen the proteins to be of variable sizes, spanning from 214 residues for ADK to 917 residues for HKC. This variability in the size of the proteins and in the type of motions presents a challenge for the method, which makes the achieved results a good indicator of its performance and its ability to scale.

As mentioned in Section 4.2.2, each iteration of the method performs a short RRT exploration. In our experiments, each RRT exploration runs until it has moved 0.3 Å C_α RMSD towards the goal. This distance is gradually reduced to 0.15 Å as the distance to the goal becomes smaller. This is because generating new valid conformations by the RRT is harder at the vicinity of the closed conformation. Exploration is also stopped after a certain number of iterations (4000 in our case) regardless of whether it has moved the required distance or not. This additional stopping condition is introduced to prevent the RRT from iterating indefinitely when it is unable to move the required distance towards the goal, which is a problem that may be solved when the normal modes are recomputed.

Once the RRT exploration stops, the conformation in the tree that is closest to to the

Protein	Residues	PDB ID$_{init}$	PDB ID$_{goal}$	ParRMSD$_{init}$	C$_\alpha$-RMSD$_{init}$
ADK	214	4ake	1ake	6.52	6.51
LAO	238	2lao	1laf	3.77	3.73
DAP	320	1dap	3dap	3.81	3.78
NS3	436	3kqk	3kql	2.75	2.75
DDT	535	1ddt	1mdt	10.93	10.96
GroEL	547	1aon	1oel	10.38	10.49
ATP	573	1m8p	1i2d	3.79	3.78
BKA	691	1cb6	1bka	4.73	4.75
UKL	876	1ukl	1qgk	6.16	6.17
HKC	917	1hkc	1hkb	2.98	3.00

Table 4.4: Details of the proteins used in the simulations. In this table, ParRMSD is the RMSD between the initial and goal conformations computed using the particles only, whereas C$_\alpha$-RMSD is the RMSD computed using the C_α atoms.

goal is identified and minimized. We have used in our experiments the AMBER software package [Case 06] for the minimization, however, any other minimization software can be used. The minimized conformation is then added to the solution path and is considered to be the starting conformation in the next RRT exploration. Normal modes for the next iteration are computed from this minimized conformation (we use the Eigen software library[1] to compute the eigenvalues and eigenvectors). Before each iteration, a quick computation is performed to find the cutoff distance that yields the best overlap value. Based on the minimized conformation, we compute overlap values using cutoffs from 14 to 18 Å and choose the best cutoff for computing the normal modes.

Results

Table 4.5 summarizes the results achieved by our method for the ten proteins. In this table, C$_\alpha$-RMSD$_{end}$ is the distance between the goal conformation and the closest conformation found by our method, which corresponds to the distance between q_{goal} and q_{close} in the last iteration of the algorithm (using the terminology of Algorithm 4.1). The table also shows the time (in hours) spent by the algorithm exploring using RRTs (Time$_{RRT}$) and the total time spent by the algorithm (Time$_{total}$). The total time includes Time$_{RRT}$ plus the time needed for running minimizations and for finding the best cutoff distance at each iteration. Finally, the number of iterations indicated in this table refers to the number of times normal modes have been computed. In all of the simulations, the time spent exploring using the RRT makes more than 90% of the total time spent by the algorithm.

[1] http://eigen.tuxfamily.org/.

Protein	C_α-RMSD$_{init}$	C_α-RMSD$_{end}$	Iterations	Time$_{RRT}$	Time$_{total}$
ADK	6.51	1.56	31	1.82	2.00
LAO	3.73	1.32	20	1.52	1.65
DAP	3.78	1.31	16	1.78	1.92
NS3	2.75	1.29	14	2.82	3.00
DDT	10.96	2.88	272	81.54	86.4
GroEL	10.49	2.79	142	40.21	42.17
ATP	3.78	1.45	30	13.46	14.16
BKA	4.75	1.96	74	29.56	31.09
UKL	6.17	2.02	80	80.61	82.62
HKC	3.00	1.64	38	37.91	39.63

Table 4.5: Performance of the method for the ten proteins.

Note that simulations were run on a single AMD Opteron 148 processor at 2.6 GHz.

In all of the performed simulations, our method has been able to find paths to conformations that are very close to the given goal conformations. All the distances between the final and goal conformations, except for DDT and GroEL, are less than or equal to 2 Å (measured using C_α-RMSD), which means that the goal can be considered as reached. Figures 4.17 to 4.26 show the final and goal conformations superimposed, and shows the superimposition of the goal and open conformations as a reference[2]. Looking at Figures 4.21 and 4.22, it is clear that the conformations found by our method for DDT and GroEL are also very close to the goal conformations, which means that the goal can be considered as reached for these proteins too. Note that the method could have reached closer conformations to the goal, however, the general strategy in our simulations was to stop when the distance to the goal reaches a very slow convergence rate.

To further analyze the time required by our method to compute conformational transitions, Table 4.6 and Figure 4.3 show the achieved results as a relationship between the number of residues and the time required by our method to compute a path that is 1Å long. Knowing this relationship is more important than knowing the exact numbers when analyzing the scalability of the method. This is because the time required to compute the path can become better or worst depending on the computers used, whereas the relationship remains the same. As seen in the figure, the scalability is linear, which is a promising property of the method. Note that the time expected for our method for proteins with more than 900 residues is better than what is shown in the figure. This is because the data point has been computed using the results of the HKC protein simulation, which

[2]The superimposition of the conformations has been performed using the software package PyMol (http://www.pymol.org/)

Protein	Residues	Time (hours) / 1Å
ADK	214	0.4
LAO	238	0.68
DAP	320	0.79
NS3	436	2.11
DDT	535	10.72
GroEL	547	5.84
ATP	573	6.74
BKA	691	11.17
UKL	876	19.96
HKC	917	28.93

Table 4.6: Relationship between the number of residues and the time (in hours) required to compute a path that is 1Å long.

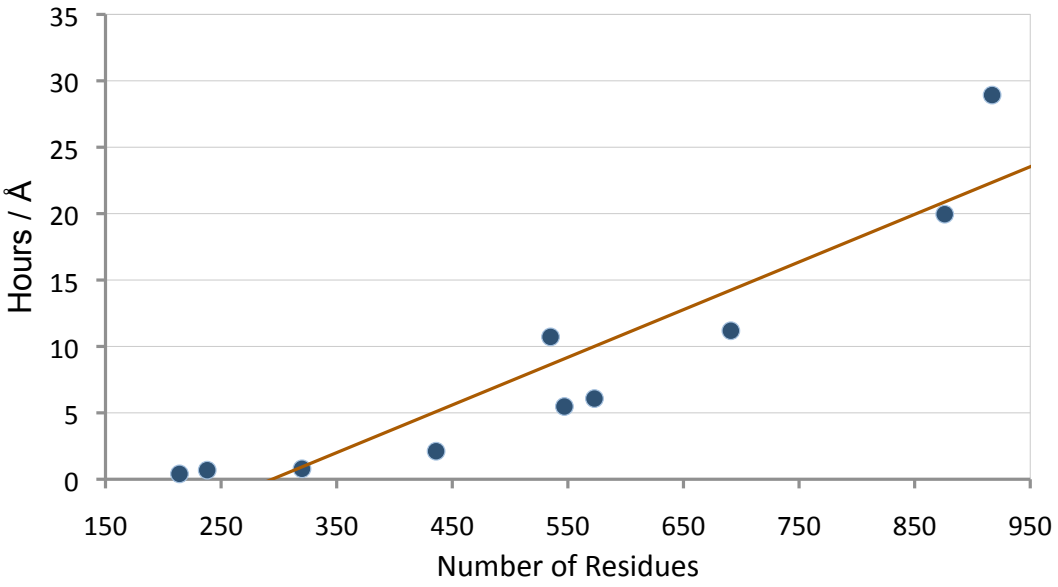

Figure 4.3: Relationship between the number of residues and the time (in hours) required to compute a path that is 1Å long.

Protein	NN Search	Collision Checking	Inverse Kinematics	Sampling (q_{rand})
ADK	57.2%	14.1%	15.0%	6.3%
LAO	51.3%	20.9%	17.0%	5.4 %
DAP	50.5%	20.6%	11.0%	12.3%
NS3	67.9%	13.4%	6.6%	8.9%
DDT	64.3%	17.1%	6.9%	9.0%
GroEL	60.4%	17.6%	8.9%	9.8%
ATP	57.3%	20.9%	6.8%	11.9%
BKA	55.1%	16.8%	6.1%	19.3%
UKL	62.9%	15.5%	4.1%	15.5%
HKC	68.9%	5.8%	3.3%	18.2%
Average	59.58%	16.27%	8.57%	11.66%

Table 4.7: The main RRT operations and the percentage of the time spent performing them.

was for a domain motion that starts at the vicinity of the closed conformation (around 3Å away). Exploring at such a distance from a compact conformation requires more time than exploring farther away (see Figure 4.6).

Table 4.7 shows the percentage of the RRT time spent by our method performing some of the most time-consuming RRT steps. An interesting observation in this table is that nearest neighbor search consumes around 60% of the computation time. This is mainly due to the brute-force nearest neighbor algorithm used in our implementation. As mentioned before, more sophisticated nearest neighbor algorithms (for e.g. [Atramentov 02]) can be used to overcome this performance bottleneck. Another possibility to improve the computational performance is to use simplified distance metrics that consume less time to perform. Examples of such metrics have been described in Section 1.1.2. Overall, these first results obtained with an unoptimized implementation could be further improved by using more sophisticated methods for the low-level operations such as nearest neighbor search and collision detection.

A Closer Look at Adenylate kinase

Adenylate Kinase (ADK) [Müller 92] is a signal transducing protein that has been studied widely (for examples see [Miyashita 03, Maragakis 05, Müller 96]). It is made of 214 amino acid residues and its structure is divided into three main domains known as: LID, CORE and NMPbind [Maragakis 05, Müller 96]. These domains are shown in Figure 4.4. The conformational transition problem we have studied is between the two conformations found in the Protein Data Bank with IDs 4ake and 1ake (open and closed respectively).

The distance between these two conformations is 6.52 Å, measured using the RMSD between the C_α atoms. It has been observed in previous studies that during this conformational transition, the LID and NMPbind domains undergo clear conformational changes, whereas the CORE domain remains almost unchanged [Maragakis 05, Müller 96]. It has also been observed that the conformational transition goes through a two-step process where the NMPbind domain moves less clearly than the LID domain at the beginning and then moves at a faster pace as the transition approaches its end [Maragakis 05].

Figure 4.4 shows the open and closed conformations along with four intermediate conformations generated by our method. As expected, the LID and NMPbind domains change significantly compared to the CORE domain. Figure 4.5 shows the displacement of the residues along the conformational transition, where darker regions represent larger displacements. In the first plot, regions around residues between 20-60 and around residues between 130-160 are clearly darker than the other parts of the plot. These regions correspond approximately to the NMPbind and LID domains respectively. It is also clear in the second plot that residues of the NMPbind domain start moving with more significance around the end of the conformational transition, whereas residues in the LID domain start at an earlier stage, which reflects the two step nature of the transition discussed earlier. These results show that the path generated by our method is in agreement with the previously found results.

We have also found four previously known intermediate conformations of the ADK protein to be very close to conformations generated by our method on the transition path. Table 4.8 shows the distance between each of these intermediate conformations and the closest conformation to it. The table also shows where the closest conformation is on the transition path. For example, 2RH5 (A) was found to be very close to the conformation generated by the first iteration, whereas 1E4Y (A) was found to be very close to the conformation generated by iteration 27. These results are in line with what has been found before in studies such as [Feng 09, Haspel 10], which further validates the agreement between the transition path generated by our method and how the ADK protein is known to move in reality.

Our method has been able to generate the transition pathway in 2 hours using 31 iterations. The time required at each iteration to minimize the closest conformation to the target and find the best overlap value was 0.35 minutes. The closest conformation to the target found by our method is 1.56 Å away from it. This distance between the reached and target conformations is very small, and therefore, the method can be considered to have reached the goal (See Figure 4.17). Figure 4.6 shows the evolution (over time) of the distance to the target and the radius of gyration. As can be seen, the method moves much more quickly when it is far from the target and slows down as it gets close to it. This is because motions are more restricted around the closed conformation than around the open one.

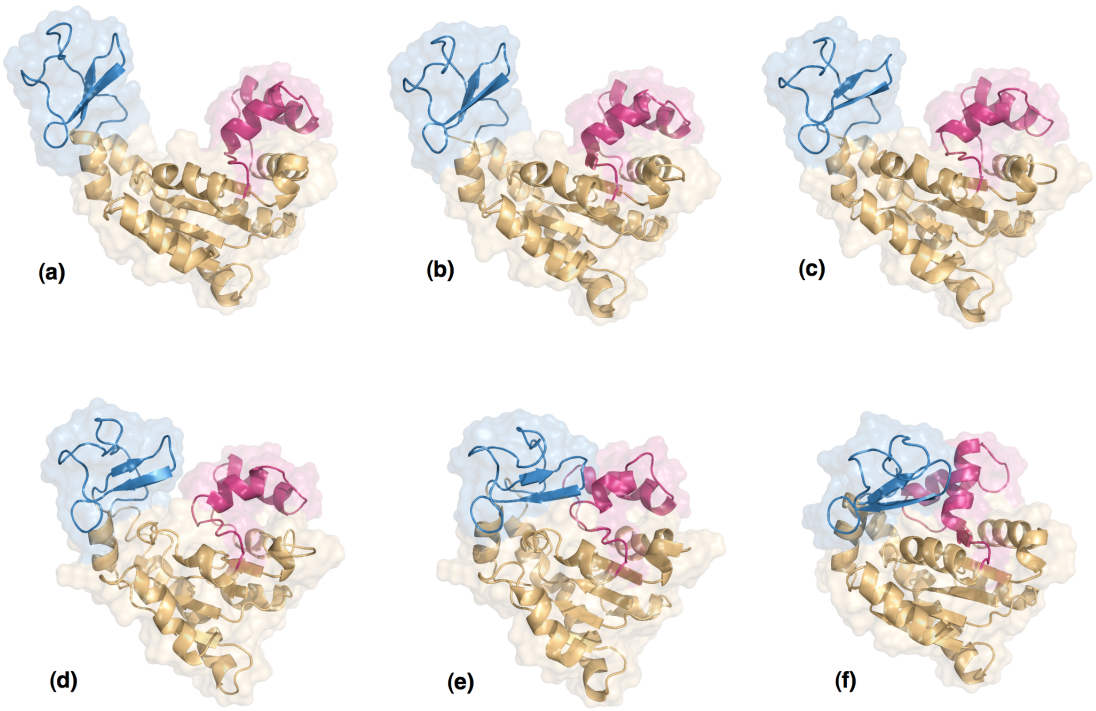

Figure 4.4: Different conformations of the ADK protein along the studied conformational transition. The LID domain is shown in blue and the NMPbind domain is shown in red. Conformations (a) and (b) are the start and goal conformations respectively, and (b), (c), (d) and (e) are conformations that have been generated by our method.

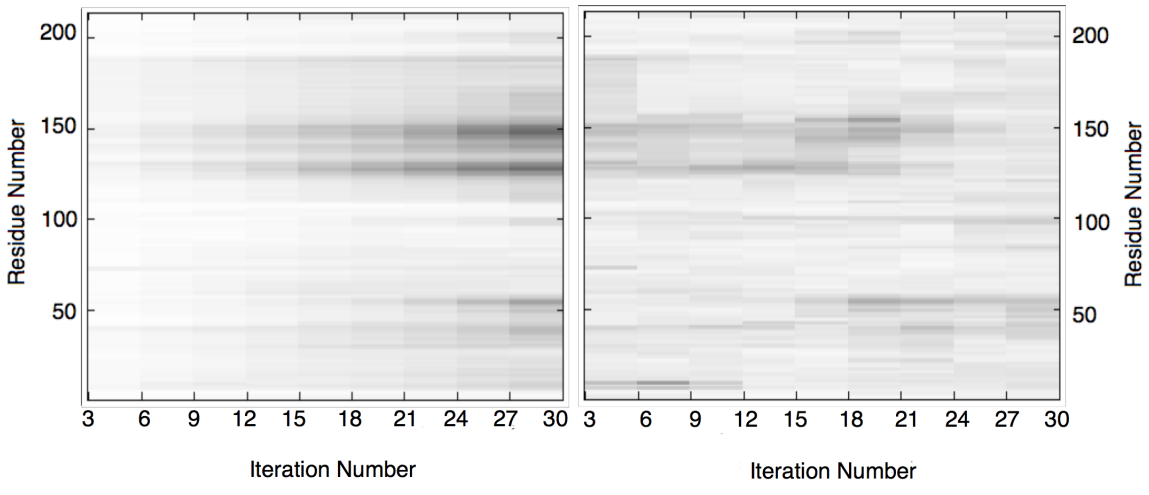

Figure 4.5: Displacement of the residues during the computed transition path. Displacements in the first plot (lef) are computed relative to the first conformation and in the second plot (right) are relative to the previous conformation. Darker regions in these plots represent larger displacements.

PDB ID	RMSD	Iteration	Percent
1DVR (A)	1.48	2	9%
2RH5 (A)	1.80	1	4%
2RH5 (B)	1.91	3	15%
1E4Y (A)	2.20	27	94%

Table 4.8: Known intermediate conformations and their distances to the closest conformations found by our method. The table also shows in which iteration the closest conformation is and where on the transition path it appears (percent).

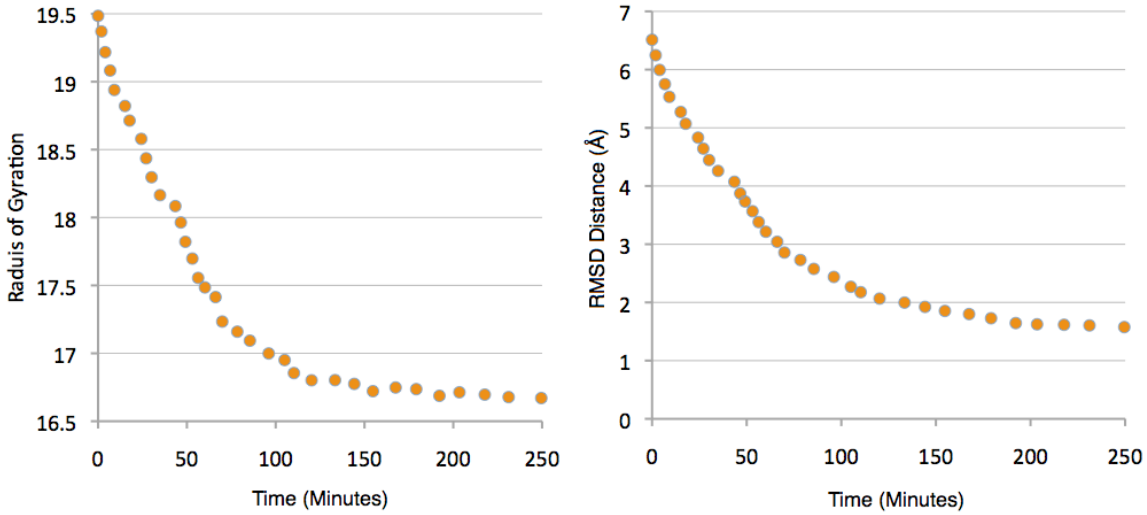

Figure 4.6: Evolution of the radius of gyration and of the RMSD distance to the goal over time.

4.4 Conclusion

This chapter has presented an efficient method for computing large amplitude motions in proteins. This method is based on the idea of combining normal modes and the RRT to speedup the exploration of the conformational space. The proposed method makes use of both the efficiency of the RRT in exploration and the ability of normal modes to locally predict motion directions. Using normal modes alone requires performing a large number of iterations and using the RRT alone wastes time in exploring irrelevant parts of the conformational space. Hence, combining the two methods allows overcoming the problems of each method. The proposed method also relies on the tripeptide-based representation of the protein, which reduces the number of computed modes and provides an accurate method for switching between the coarse-grained model and the full atom model.

Performed experiments have shown that computing normal modes of a protein using its

simplified particle-set instead of the C_α atoms does not lead to a degradation in the ability to predict motion directions. Results also have shown that the proposed method is able to compute paths for conformational transitions of different lengths in proteins of different sizes and topologies. The performance of the method scales linearly with the number of residues. Using a single AMD Opteron 148 processor at 2.6 GHz, studying transitions takes a few hours in small proteins and a few days in large ones depending on the length of the computed path. Note however that computing times have been shown for a first unoptimized implementation of the method. Improvements in time-consuming functions such as nearest neighbor search could significantly speed-up computations. Analysis of the conformational transition in the ADK protein by our method shows also that it is able to produce paths that are consistent with previously found results.

An interesting extension of the method that can be investigated is the prediction of unknown candidate conformations. This problem is more challenging than the one studied in this chapter since the goal conformation is missing. However, using the normal modes, the RRT may be able to identify one or more candidate target conformations. Another interesting extension to test is the use of a bi-directional RRT [Kuffner Jr 00] that starts two trees rooted at each of the open and closed conformations. It is also worth to test the effect of using a parallelized version of the RRT as in [Devaurs 11], which could improve the overall performance of the method. Finally, a possible direction to investigate is the use of a Mitropolis-like test (as in [Jaillet 10, Jaillet 11]) to accept or reject new conformations in the tree instead of the purely-geometric test that we currently use.

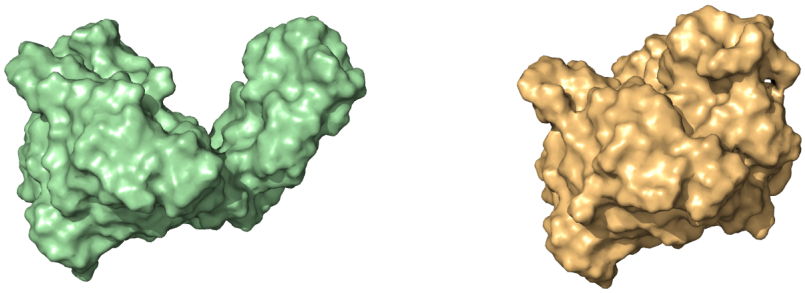

Figure 4.7: ADK: 4ake (left) and 1ake (right)

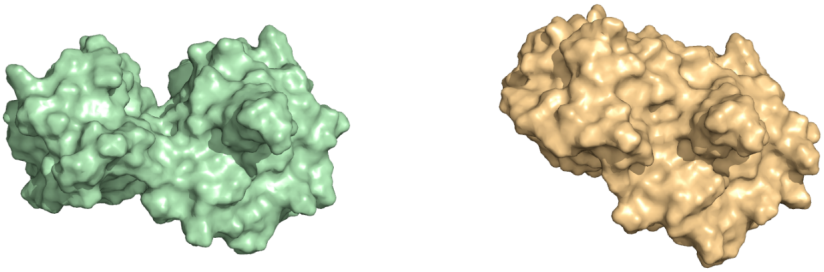

Figure 4.8: LAO: 2lao (left) and 1laf (right)

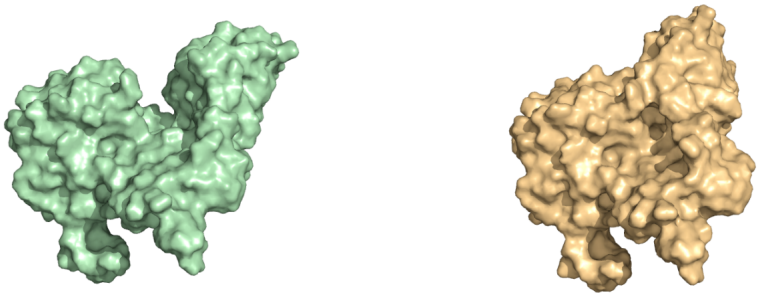

Figure 4.9: DAP: 1dap (left) and 3dap (right)

Figure 4.10: NS3: 3kqk (left) and 3kql (right)

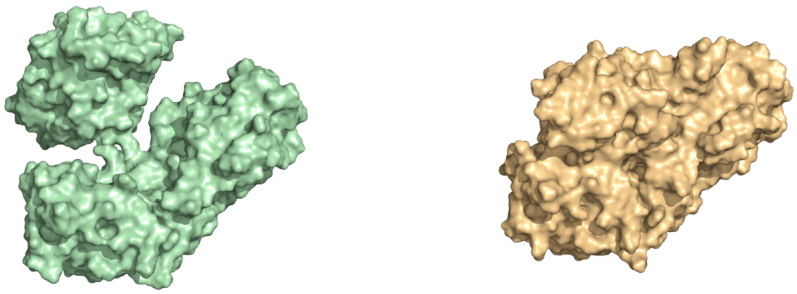

Figure 4.11: DDT: 1ddt (left) and 1mdt (right)

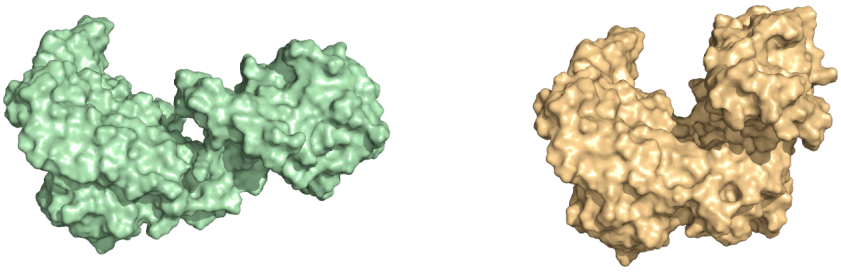

Figure 4.12: GroEL: 1aon (left) and 1oel (right)

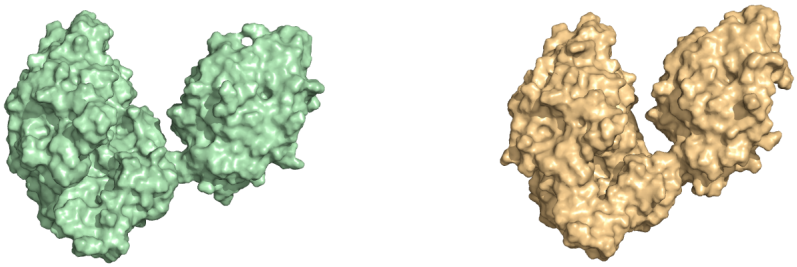

Figure 4.13: ATP: 1m8p (left) and 1i2d (right)

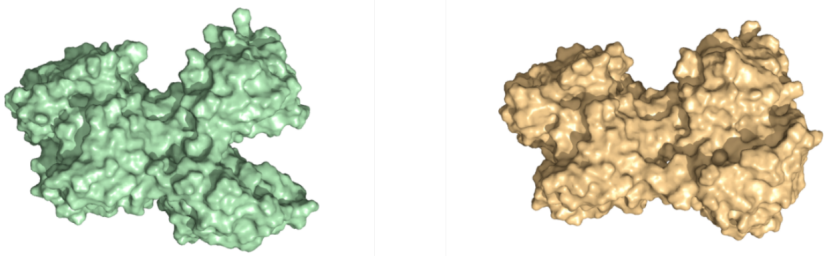

Figure 4.14: BKA: 1cb6 (left) and 1bka (right)

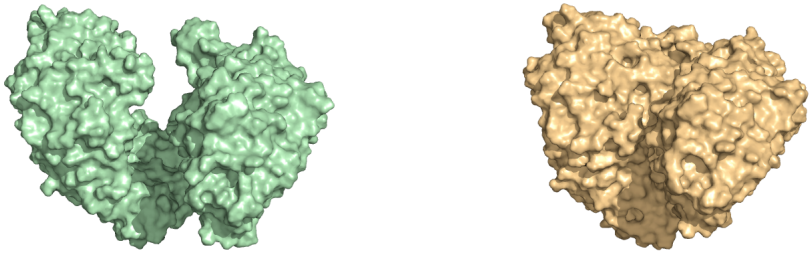

Figure 4.15: UKL: 1ukl (left) and 1qgk (right)

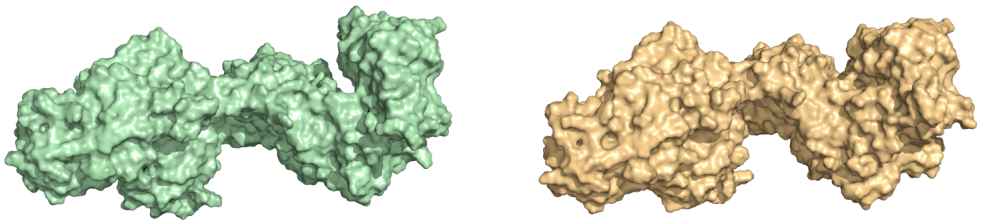

Figure 4.16: HKC: 1hkc (left) and 1hkb (right)

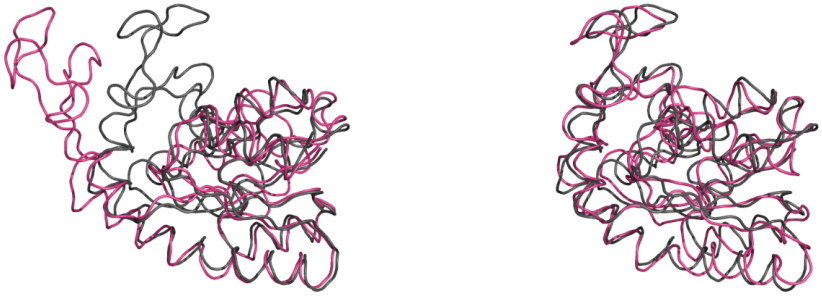

Figure 4.17: ADK: q_{init} (left), final conformation (right) and q_{goal} (in black).

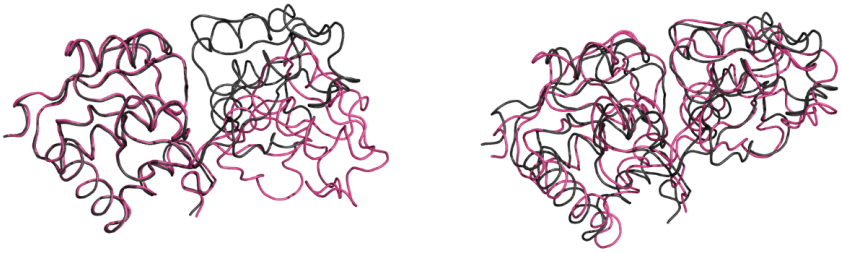

Figure 4.18: LAO: q_{init} (left), final conformation (right) and q_{goal} (in black).

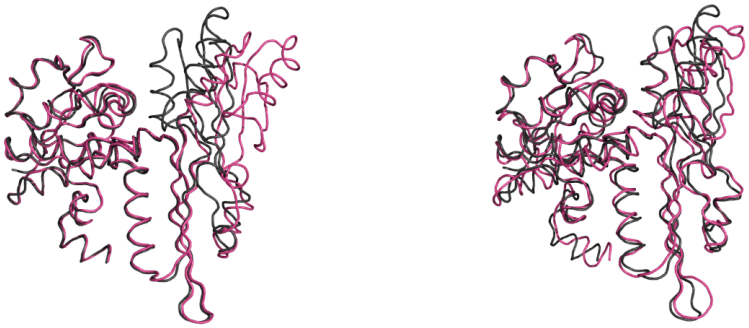

Figure 4.19: DAP: q_{init} (left), final conformation (right) and q_{goal} (in black).

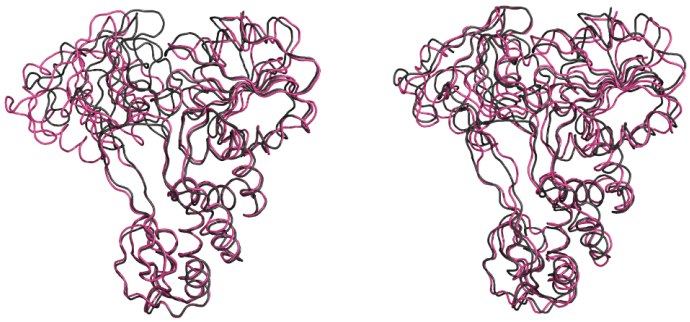

Figure 4.20: NS3: q_{init} (left), final conformation (right) and q_{goal} (in black).

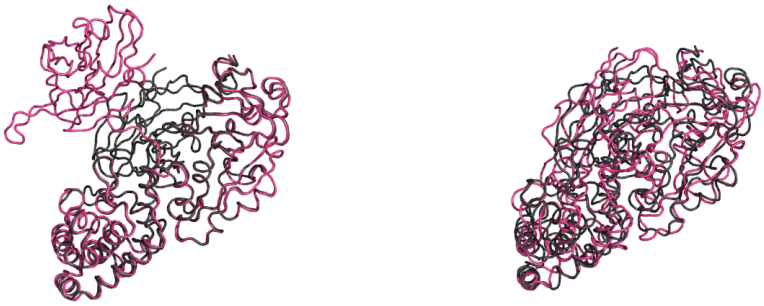

Figure 4.21: DDT: q_{init} (left), final conformation (right) and q_{goal} (in black).

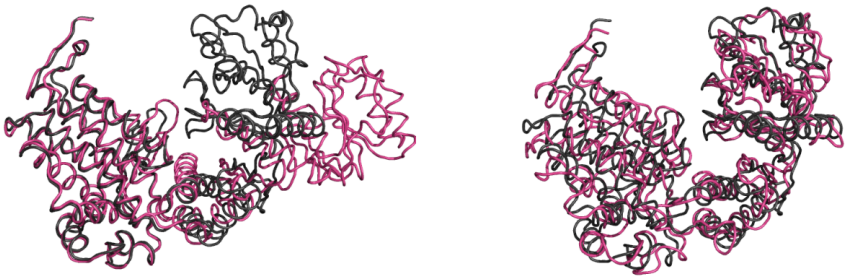

Figure 4.22: GroEL: q_{init} (left), final conformation (right) and q_{goal} (in black).

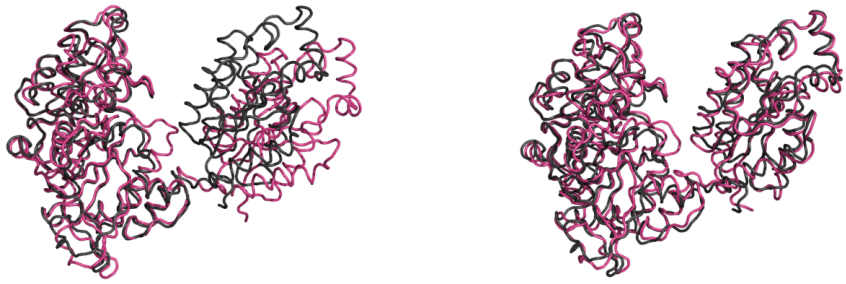

Figure 4.23: ATP: q_{init} (left), final conformation (right) and q_{goal} (in black).

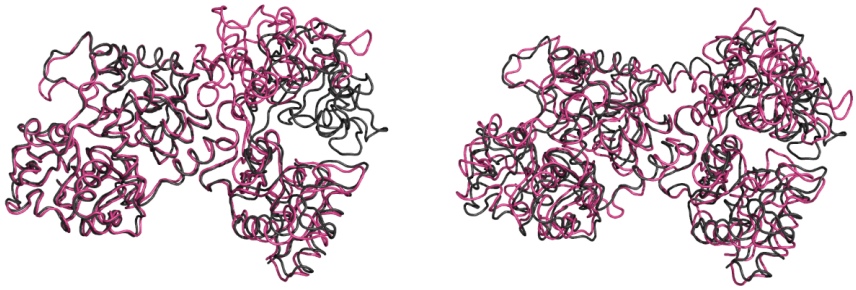

Figure 4.24: BKA: q_{init} (left), final conformation (right) and q_{goal} (in black).

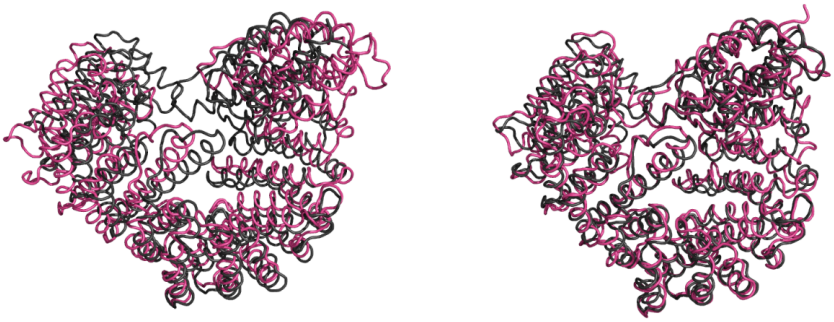

Figure 4.25: UKL: q_{init} (left), final conformation (right) and q_{goal} (in black).

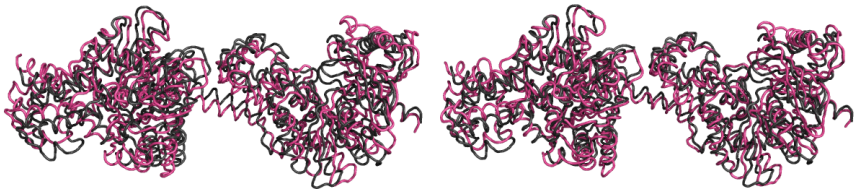

Figure 4.26: HKC: q_{init} (left), final conformation (right) and q_{goal} (in black).

Conclusions

We have presented in this thesis a robotics-inspired approach for protein modeling, called the tripeptide-based model, and have shown how it can be used to enhance molecular simulations. The modeling approach provides a high-level (coarse-grained) representation based on a mechanistic subdivision of the protein into short kinematic chains called tripeptides. At the same time, it provides an accurate method for generating low-level (full-atom) details using 6R inverse kinematics whenever needed. The advantage of this kind of modeling, as shown in this thesis, is that it provides a unified method for implementing a variety of simulation algorithms that are able to treat proteins efficiently using the coarse-grained representation, but without loosing full-atom details.

In addition to the presentation of the tripeptide-based model, we have shown two different applications for its use in enhancing molecular simulations. In the first application, we have used the tripeptide-based model to implement new Monte Carlo move classes as well as several others that have been proposed in the last decades for improving protein backbone sampling. The flexibility of the tripeptide-based model enabled us also to easily combine these move classes into a mixed sampling strategy that alternates sampling between them. Simulations performed with two proteins of different sizes and topologies have validated the applicability of this approach. The performed simulations have also shown that the mixed sampling strategy provides a clear performance gain over the other implemented move classes. The mixed strategy was able to explore conformations that are better than the conformations explored by the other move classes, in terms of energy and structural variability, without demanding high computational resources.

In the second application, we have presented a motion planning inspired method for studying large amplitude motions in proteins. This method combines the tripeptide-based model, RRT and normal mode analysis to explore efficiently conformational transitions in proteins with more than a thousand degrees of freedom. Although the RRT is known to quickly explore high dimensional spaces, it can waste considerable time exploring parts of the space that are not directly relevant to the studied conformational transition. Nev-

ertheless, information given by the normal modes, allowed us to overcome this limitation by biasing the exploration of the RRT towards the more relevant parts of the space. Moreover, the use of the tripeptide-based model reduced the number of computed normal modes, which enhanced the overall performance of the algorithm. It also provided an accurate method for keeping track of the full atom details during the exploration. Simulations performed using our method have shown that elastic networks built using the tripeptide-based model can predict motion directions with an accuracy that is comparable to the directions computed using C_α elastic networks. Simulations have also shown that the introduced method can compute transition paths between conformations in proteins of different sizes and topologies and provides a performance that scales linearly with the number of residues. Detailed analysis of the computed path for the ADK protein has also shown that the method produces paths that are in agreement with results found by other, more expensive methods.

Future Work

The presented work on the application of the tripeptide-based model for enhancing Monte Carlo simulations can be extended and further investigated. First of all, new move classes that are based on the perturbation of a particle or a group of particles need to be tested and compared in more detail with the other available move classes. Questions that still need more precise answers concern the type of protein topologies and molecular simulation problems that are more suited to these move classes, and which variants of these move classes provide better performance. Similarly, further tests need to be performed in order to identify optimal combinations of move classes or probabilities of usage in the mixed strategy for the different protein topologies and molecular simulation problems.

The NMA-guided RRT was used in this thesis to find a transition path between two known conformations. However, applications of this method surpass this application, as it can be used to study other types of problems that require an exploration of the conformational space. For example, our method can possibly be used to predict most probable conformations that the protein can reach from a given conformation. Other possible research directions have been highlighted also in the conclusion of Chapter 4, such as investigating the use of parallelized RRTs [Devaurs 11], bi-directional RRTs [Kuffner Jr 00] and Transition-based RRTs [Jaillet 10, Jaillet 11]. These RRT variants have been shown to improve the performance of the RRT, which makes them candidates for improving the performance of our method too.

The methods presented in this thesis are focused on the flexibility of the protein backbone, while side-chain flexibility has been treated using simplistic approaches such as random sampling or local energy minimization. More explicit methods for treating side-chains should be explored in order to allow more control over them during simulations.

Another issue that would need further investigation is the use of energy models that are more suitable to the proposed tripeptide-based model. A multi-level model could be developed to provide coarse-grained (computationally cheap) energy evaluations when working with the high-level representation of our tripeptide-based model, and a more accurate (computationally expensive) energy evaluations when dealing with the all-atom representation.

Our goal in the short future is to investigate introducing more abstraction to the tripeptide-based model. Currently, the model includes only two levels: the simplified particle-based representation and the full-atom representation, which can be a limitation, especially when dealing with very large protein systems. It is possible, for example, to add another layer that joins several consecutive tripeptides into one fragment. This can be useful in treating large proteins with limited flexibility. It is also possible to subdivide the protein into variable-length fragments depending on the flexibility in the different parts of the protein. Secondary structures, for example, are known to be less flexible than protein loops, which is why it is reasonable to use fragments that include more than three residues there.

We also plan to investigate the use of the tripeptide-based model for studying molecular interactions, where considering the full flexibility of the interacting proteins poses a real challenge. Using the tripeptide-based modeling approach, the flexibility of the proteins can be treated using the particles instead of having to deal with all the degrees of freedom. At the same time, the full-atom model of the proteins can be generated whenever necessary. On the other hand, molecular interactions induce deformations that are different from those that naturally occur in response to internal forces. One way to study interactions-induced deformations is to make use of Static Modes [Brut 09], which provide motion directions that occur in response to the application of external forces. Hence, Static Modes could replace normal modes in the method discussed in Chapter 4, in order to guide the exploration towards potential binding conformations.

Résumé étendu

Introduction

Les simulations numériques sont largement utilisées aujourd'hui pour modéliser les biomolécules, imiter leur comportement, et avoir un aperçu de leurs propriétés physico-chimiques et de leurs fonctions biologiques. En effet, un domaine entièrement dédié à ce genre de simulation existe sous le nom de biologie structurale computationnelle.

Les méthodes computationnelles ont été essentiellement développées pour compléter les méthodes expérimentales. Par exemple, la dynamique moléculaire (MD) [Rapaport 07] et la méthode de Monte Carlo (MC) [Landau 05] sont largement utilisées pour étudier les propriétés thermodynamiques et l'activité des protéines à partir d'une structure initiale déterminée par cristallographie aux rayons X [Woolfson 97] ou par résonance magnétique nucléaire (NMR) [Cavanagh 06]. La complémentarité entre les méthodes expérimentales et les méthodes informatiques ou computationnelles peut également être exploitée dans l'autre sens, puisque les simulations peuvent être améliorées en utilisant des données expérimentales. Une illustration intéressante de cette complémentarité est l'utilisation de déplacements chimiques de NMR pour restreindre les simulations [Robustelli 10].

Certaines méthodes computationnelles vont plus loin dans le but de remplacer les méthodes expérimentales. Par exemple, certaines méthodes informatiques peuvent être utilisées pour déterminer la structure des protéines sans avoir d'information expérimentale antérieure [Bonneau 01]. Des méthodes sont également disponibles pour évaluer les interactions moléculaires (molecular docking) [Lengauer 96], et pour comprendre comment les protéines passent d'un état de pelote aléatoire vers leur structure native (protein folding) [Pain 00]. Néanmoins, l'état actuel de ces méthodes informatiques est encore loin de leur permettre de fournir des résultats tout-à-fait précis et fiables dans tous les cas. Les exemples les plus complexes parmi les problèmes mentionnés ci-dessus restent hors de portée des méthodes de l'état de l'art. Par exemple, la puissance computationnelle actuelle permet l'exécution de simulations MD couvrant seulement quelques microsecon-

des de temps physique. Ceci est bien sûr insuffisant car les mouvements moléculaires, lors de certains processus tels que le repliement de protéine (Protein Folding), peuvent se produire sur une durée de plusieurs secondes [Muñoz 08]. Les méthodes MC souffrent aussi de lacunes dans leur exploration et leur échantillonnage de l'espace conformationnel des protéines, qui est un paysage accidenté avec de nombreux minima locaux. Les méthodes MC ont tendance à se retrouver bloquées dans ces minima locaux et à perdre un temps considérable à essayer de s'en échapper.

De ce fait, des recherches actives se concentrent actuellement sur l'amélioration des techniques de simulation (voir par exemple [Sugita 99, Marinari 92, Laio 02, Shaw 10]) et sur la production de méthodes alternatives. Cette thèse s'inscrit dans une classe particulière parmi ces méthodes alternatives : celles qui sont inspirées par le domaine de la planification de mouvement en robotique. Les méthodes inspirées par la robotique ont été introduites récemment pour simuler les mouvements de protéines et étudier des problèmes comme le repliement des protéines (Protein Folding) et les interactions entre protéines et ligands. Ils sont principalement basés sur les algorithmes de planification de mouvement par échantillonnage [LaValle 06, Choset 05, Tsianos 07], qui se sont révélés être de puissants outils pour résoudre les problèmes faisant intervenir des espaces de haute dimension.

Bien que les deux domaines (robotique et simulation moléculaire) semblent très éloignés au premier abord, une comparaison plus approfondie révèle de nombreuses similarités en termes de formulation des problèmes abordés. Dans un article présentant l'état de l'art à ses débuts [Parsons 94], Parsons et Canny ont montré que plusieurs problèmes étudiés dans le domaine de la biologie structurale computationnelle sont effectivement des problèmes géométriques qui ont leurs équivalents dans le domaine robotiques. Cette similarité est due principalement au fait que le mouvement joue un rôle central, que ce soit pour les robots ou les protéines. En effet, les mouvements moléculaires font partie intégrante des processus biologiques dans lesquels les protéines sont impliquées, comme par exemple la catalyse et la transmission du signal. Le fait de comprendre comment les protéines se déplacent conduit à la compréhension de ces processus, ainsi qu'à la compréhension de leurs dysfonctionnements et de leur contribution à des maladies telles que la maladie de la vache folle ou la maladie d'Alzheimer [Selkoe 03].

Dans cette thèse, nous présentons une approche de modélisation mécanistique des protéines et nous montrons comment elle peut être utilisée pour améliorer les simulations moléculaires. Cette approche de modélisation utilise des notions de robotique permettant un traitement haut niveau (coarse grained) des molécules, sans perdre les détails au niveau atomique (all-atom). Nous montrons comment cette approche de modélisation peut être utilisée pour mettre en œuvre des classes de mouvements de Monte Carlo, et comment elle peut conduire à une amélioration de l'échantillonnage global de l'espace conformationnel moléculaire. Nous proposons également, en nous basant sur cette approche

de modélisation, une approche de planification du mouvement combinée avec la méthode d'analyse en modes normaux (Normal Mode Analysis NMA) [Cui 06] pour étudier les mouvements de grande amplitude dans les protéines. L'utilisation de l'approche de modélisation mécanistique avec la méthode RRT de planification de mouvement [LaValle 01a] et l'analyse en modes normaux NMA offre un clair gain en performance, ce qui nous permet de présenter des résultats de simulations de transitions conformationnelles pour des protéines contenant jusqu'à mille résidus. Outre la contribution méthodologique, cette thèse propose également une étude exhaustive de l'utilisation des algorithmes de planification de mouvement dans les simulations moléculaires. A notre connaissance, la littérature ne contient pas une telle étude, bien qu'elle puisse être utile aussi bien pour les roboticiens que pour les biologistes désireux de travailler dans ce domaine.

La thèse est organisée autour de ces contributions comme suit. Le chapitre 1 est consacré à passer en revue et à analyzer l'utilisation des méthodes inspirées par la planification de mouvement dans les simulations moléculaires. Ensuite, le chapitre 2 présente les détails de l'approche mécanistique de modélisation des protéines, qui sert de base pour les méthodes présentées dans les deux chapitres suivants. Le chapitre 3 est consacré aux applications de cette approche de modélisation dans les simulations de Monte Carlo. Puis, le chapitre 4 présente la méthode combinée RRT-NMA ainsi que l'étude de simulations de transitions conformationnelles dans des protéines de différentes tailles. Enfin, la thèse se termine par une conclusion et une discussion des directions de recherche futures.

Un bref résumé de chaque chapitre

Chapitre 1: Algorithmes de planification de mouvement pour les simulations moléculaires

Ce chapitre présente un état de l'art concernant les algorithmes de planification de mouvement appliqués à la modélisation moléculaire ainsi qu'à la simulation. Sont discutés dans ce qui suit, aussi bien les aspects algorithmiques qu'applicatifs. Une attention spéciale a été portée aux questions concernant l'extension des algorithmes de planification de mouvement de la robotique au domaine moléculaire. D'un point de vue algorithmique, le chapitre donne un aperçu général des différents algorithmes de planification de mouvement par échantillonnage proposés dans ce contexte. D'un point de vue applicatif, le chapitre traite les problèmes liés au repliement des protéines, aux transitions conformationnelles ainsi qu'aux interactions de type "protéine-ligand".

A notre connaissance, les algorithmes de modélisation moléculaire et de simulation inspirés par la planification de mouvement sont relativement nouveaux. Par conséquent, il n'existe pas d'étude dédiée à ce sujet. De ce fait, l'objectif de ce chapitre est double. Premièrement, en expliquant les concepts liés à la planification de mouvement ainsi que

les applications qui en sont faites dans le domaine moléculaire, ce chapitre pose les bases des chapitres suivants. Deuxièmement, ce chapitre tente d'enrichir le peu de littérature existant dans le domaine par une étude exhaustive étayée par une analyse des usages faits des algorithmes de planification de mouvement en simulation de molécules. Pour les lecteurs apparentés à la communauté des biologistes, cette étude peut jouer le rôle d'une introduction aux méthodes inspirées par la robotique et utilisées dans le domaine de la biologie structurale. Inversement, pour les lecteurs apparentés à la communauté des roboticiens, cette étude peut jouer le rôle de catalyseur pour l'étude de nouvelles opportunités d'applications liées à la biologie structurale, et inciter de nouveaux développements ainsi que de nouvelles adaptations et améliorations d'algorithmes pour une résolution plus fine de problèmes impliquant de larges espaces multidimensionnels.

Des travaux présentés dans ce chapitre, il ressort que les algorithmes inspirés par la robotique sont des pistes prometteuses, dès lors qu'ils sont combinés à des techniques plus conventionnelles de calcul en biologie structurale. Leur atout majeur réside principalement dans leur efficacité à explorer des espaces d'une grande complexité. Comparés à des méthodes plus classiques telles que MC, les algorithmes de planification de mouvement par échantillonnage ne requièrent que peu d'itérations pour trouver des chemins de transitions conformationnelles ou encore pour obtenir un ensemble représentatif d'états conformationnels. De surcroit, à l'inverse des simulations de type MD, ces algorithmes n'ont pas besoin d'un champ de force pour guider l'exploration. Il en découle que différents types de données, y compris de simples modèles géométriques, peuvent être utilisés pour contraindre ou influencer l'exploration. L'utilisation de modèles simples permet l'obtention de méthodes de calcul générales et rapides, capables d'explorer de larges régions de l'espace conformationnel. Les résultats d'une telle exploration peuvent être par la suite analysés et affinés en utilisant un modèle énergétique plus adéquat.

Les méthodes inspirées par la planification de mouvement pour la simulation de molécules en sont encore à leurs balbutiements. Il est nécessaire d'améliorer ces dernières et de les valider sur des systèmes à plus large échelle. D'autres tests sur des applications réelles, menés conjointement avec des méthodes expérimentales, permettront d'améliorer ces méthodes de calcul. Des travaux supplémentaires utilisant les concepts de la physique statistique sont également nécessaires pour la caractérisation des résultats fournis par ces algorithmes.

Les classes de problèmes auquelles les méthodes de planification de mouvement ont été appliquées en biologie structurale sont très limitées : il s'agit essentiellement de la flexibilité des protéines/RNA et des interactions de type "protéine-ligand". Néanmoins, nous pensons que le champ d'application de ces méthodes est plus large et que d'autres applications peuvent être investiguées. A titre d'exemple, d'autres problèmes en biologie structurale, qui pourraient être adressés, concernent la prédiction des interactions protéine-protéine ou encore l'analyse conformationnelle de grands assemblages moléculaires.

Chapitre 2: Un modèle mécanistique pour les protéines

Ce chapitre présente une approche mécanistique pour la modélisation des protéines. L'idée de cette approche s'articule autour d'une décomposition en fragments qui peuvent être traités comme des chaînes cinématiques courtes. Une telle décomposition produit une représentation multi-niveaux de la protéine. Celle-ci induit une approche de type gros-grains et permet d'améliorer significativement les performances. Il n'empêche que les détails du niveau atomique ne sont pas perdus et peuvent à tout moment être générés à partir de la représentation de haut niveau. Ce type de modélisation fournit une approche unifiée pour l'implémentation d'une grande variété de techniques de simulation, aussi bien existantes que nouvelles. Cela sera le propos des deux prochains chapitres.

La modélisation adoptée subdivise la chaine polypeptidique en fragments contenant chacun exactement trois résidus d'acides aminés (constituant ainsi un tripeptide). Dans notre modèle, chaque tripeptide peut être assimilé à un bras articulé avec six articulations rotoïdes (c'est-à-dire avec six degrés de liberté). En effet, chaque tripeptide possède trois résidus d'acides aminés, et chaque résidu s'articule autour de deux angles diédraux mobiles (ψ et ϕ). Nous associons un repère cartésien à chaque groupe d'atomes dans le tripeptide. En outre, nous estampillons différemment les repères importants pour notre modèle. Ces repères sont le premier et le dernier dans chaque tripeptide. Ils correspondent respectivement à la base ainsi qu'au dernier repère de notre bras articulé. Les repères de base sont appelés *particules* (orientées). Au niveau de chaque tripeptide, le dernier repère peut être calculé à partir de la "particule" du tripeptide suivant, et ce en appliquant des transformations constantes. Cela est rendu possible par le fait que les tripeptides sont reliés par des liaisons peptidiques rigides. Nous nous référons au modèle de la protéine n'incluant que les repères des particules en utilisant le terme *simplified particle-set model*.

La figure 5 illustre l'application du modèle proposé à un domaine SH3 (PDB ID: 1V1C). La Figure 5.a représente le modèle de la protéine incluant le squelette dans le modèle de la surface de la protéine. La Figure 5.b illustre la trace du squelette de la protéine avec les repères correspondants aux particules. Les Figures 5.c and 5.d représentent respectivement les modèles chimiques et mécanistiques du squelette d'un tripeptide.

L'idée principale apportée par cette modélisation est de permettre l'échantillonnage, la déformation, et plus généralement tout traitement de la protéine, en utilisant uniquement le modèle simplifié, plutôt que de manipuler tous les atomes. Etant donnée une configuration spatiale, générer les valeurs des angles diédraux correspondants à chaque tripeptide, et par conséquent pour le modèle complet (all-atom), peut être effectué à l'aide de la *cinématique inverse*. La raison pour laquelle nous avons subdivisé la protéine en tripeptides avec six angles diédraux repose sur le fait que le tripeptide est le plus court fragment offrant une mobilité complète du dernier repère par rapport au repère de base.

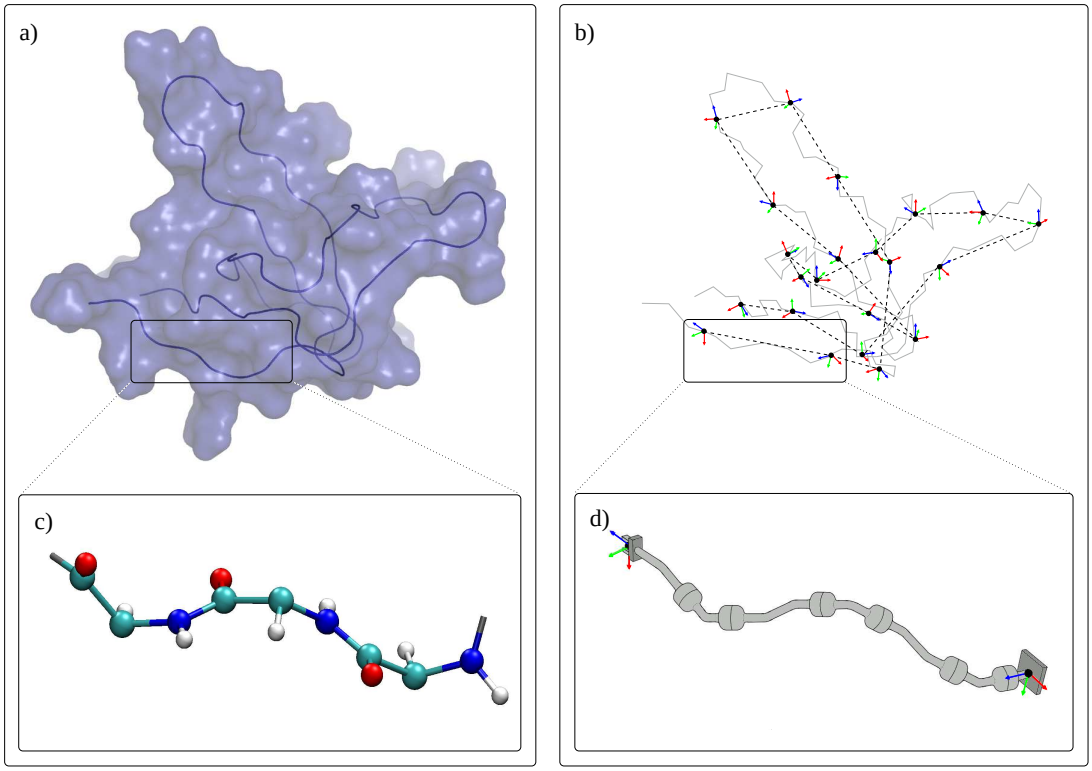

Figure 5: Une illustration de l'approche proposée. Les tripeptides, constitués de trois résidus d'acides aminés, sont traités comme des chaînes cinématiques similaires à des robots manipulateurs.

En d'autres termes, étant donné le repère de base, le dernier repère requiert au minimum six angles diédraux afin de pouvoir balayer toutes les positions possibles.

Chapitre 3: Amélioration de la méthode de Monte Carlo

Ce chapitre présente un exemple d'application de l'utilisation du modèle basé sur les tripeptides, présenté dans le chapitre précédent. Il montre comment ce modèle peut être utilisé pour faciliter l'implémentation de classes de mouvement de Monte Carlo aussi bien classiques que nouveaux. L'idée principale est de perturber la pose (position et orientation) des particules, puis d'adapter la conformation des tripeptides afin de maintenir l'intégrité de la chaîne moléculaire tout en conservant la géométrie locale des liaisons (i.e. une longueur constante des liaisons et des angles de liaison constants). Plusieurs stratégies peuvent être considérées pour perturber la pose des particules. Le nombre de particules sélectionnées pour la perturbation, ainsi que la corrélation ou non-corrélation de la direction du mouvement de plusieurs particules, conduisent à différentes classes de mouvement. Ce qui suit sont des exemples de classes de mouvement qui peuvent être implémentées en utilisant le modèle tripeptidique:

- *Déplacement d'une particule* : La classe de mouvement la plus simple est la perturbation d'une particule (i.e. la perturbation de sa pose). Une telle perturbation exige d'ajuster la conformation des deux tripeptides dont les extrémités et les repères de base définissent la particule. Cela peut être obtenu par la résolution de la cinématique inverse pour chacun des deux tripeptides. Par conséquent, ce mouvement introduit des modifications au niveau d'exactement douze angles dièdres consécutifs dans le squelette de la protéine.

- *Déplacement d'un fragment flexible* : Une simple extension à la classe de mouvement d'une particule est de perturber un certain nombre de particules consécutives au lieu d'une seule. Notez que la perturbation de n particules dans des directions aléatoires nécessite de résoudre une cinématique inverse $n+1$ fois afin d'ajuster la conformation de tous les tripeptides qui sont liés aux particules perturbées.

- *Déplacement d'un corps rigide en bloc* : Contrairement aux mouvements de fragments flexibles qui perturbent de façon indépendante n particules, cette classe de mouvement perturbe n particules consécutives en même temps, comme un seul corps rigide. En d'autres termes, les n particules ont été translatées et/ou mises en rotation autour d'un axe arbitraire tout en conservant leurs positions et orientations relatives. Par conséquent, les conformations des tripeptides entre ces particules ne changent pas. Cependant, les conformations du tripeptide se situant avant la première particule et du tripeptide se situant après la dernière particule doivent être ajustées en utilisant la cinématique inverse.

- *Classes de mouvement mixte* : Un des avantages du modèle tripeptidique proposé est qu'il fournit une approche unifiée pour la mise en œuvre de plusieurs classes de mouvement. Ceci nous permet de créer facilement une stratégie d'échantillonnage de haut niveau qui fait usage de plus d'une classe de mouvement. L'utilisation de plus d'une classe de mouvement introduit plus de variabilité au niveau du mouvement échantillonné, ce qui conduit à une meilleure couverture de l'espace conformationnel.

Nous avons effectué des tests sur deux protéines (formées de 68 et 77 résidus respectivement) afin d'évaluer les classes de mouvement implémentées en utilisant le modèle tripeptidique, et afin de les comparer à deux classes de mouvements plus traditionnelles ([Lal 69] et [Dodd 93]). Après cent mille étapes d'équilibration, des simulations MC ont été exécutées en utilisant deux pas différents, et ont été arrêtées après qu'un million de conformations soient acceptées. Les simulations réalisées avec la stratégie *mixte* ont montré un profil de distance moyenne à la conformation initiale qui est supérieur à ceux de toutes les autres classes de mouvement. Cela signifie que cette stratégie peut, en moyenne, visiter des conformations plus éloignées que n'importe laquelle des quatre autres classes de mouvement. Les simulations ont également montré que la stratégie *mixte* est capable de

maintenir un profil d'énergie moyenne qui est inférieur à ceux de toutes les autres classes de mouvement. Cette performance principale est une conséquence directe de la diversité des fluctuations structurelles obtenues par cette stratégie d'échantillonnage, puisqu'elle alterne entre quatre classes de mouvement différentes.

Chapitre 4: Exploration des transitions conformationnelles

Ce chapitre introduit une nouvelle méthode pour explorer l'espace conformationnel des protéines. La méthode est basée sur notre représentation tripeptidique des protéines, et applique une combinaison de l'algorithme RRT [LaValle 01a, LaValle 01b] et de l'analyse en mode normal (NMA) [Cui 06]. Cette méthode est particulièrement utile pour analyser les transitions entre différentes conformations d'une protéine, en particulier celles qui impliquent des mouvements de domaine.

L'étude des transitions conformationnelles dans les protéines est importante pour comprendre leurs fonctions biologiques, parce que ces mouvements sont généralement liés à la capacité de la protéine d'interagir avec d'autres molécules. Toutefois, la collecte de ce type d'information dynamique à l'échelle atomique est difficile en utilisant des méthodes expérimentales. Par conséquent, les méthodes de calcul comme la dynamique moléculaire et Monte Carlo sont le plus couramment utilisées. Néanmoins, ces méthodes souffrent également de problèmes d'efficacité lorsqu'elles sont utilisées pour évaluer les changements conformationnels de grande amplitude.

Dans ce contexte, nous proposons une méthode de calcul qui étend les méthodes introduites dans [Cortés 05b, Kirillova 08]. Ces méthodes utilisent un RRT pour accélérer l'exploration de l'espace conformationnel, et donc permettre la simulation de mouvements de grande amplitude dans les protéines, avec peu de ressources de calcul. La méthode introduite dans [Kirillova 08] va encore une étape plus loin et utilise la méthode d'analyse en mode normal pour orienter la recherche du RRT vers les régions d'énergie favorable, ce qui permet d'étudier des problèmes avec un nombre de dimensions encore plus élevé. Cette idée de biaiser l'exploration du RRT en utilisant les modes normaux est ancrée dans des travaux tels que [Brooks 85, Hinsen 98, Tama 01, Alexandrov 05], qui montrent la capacité des modes normaux à prédire la direction des changements conformationnels collectifs (comme les mouvements de domaines) dans les macromolécules. Toutefois, comme les modes normaux fournissent des prévisions locales et pas des trajectoires conformationnelles complètes, des méthodes itératives ont été introduites qui effectuent des déplacements courts et recalculent les modes normaux à chaque étape [Mouawad 96, Miyashita 03, Jeong 06]. De telles méthodes nécessitent un grand nombre d'itérations pour calculer les grandes transitions conformationnelles, ce qui peut être évité en utilisant RRT, comme cela a été montré dans [Kirillova 08].

La méthode proposée ici utilise également les modes normaux pour biaiser la recherche

du RRT. Cependant, la principale différence avec [Kirillova 08] est que notre méthode est basée sur notre modèle tripeptidique. Un tel changement, en apparence mineur, a néanmoins des conséquences importantes. En utilisant ce modèle, le nombre de modes normaux par protéine est réduit d'un facteur au moins trois, ce qui diminue considérablement le temps nécessaire pour les calculer. Un autre avantage d'utiliser le modèle tripeptidique est qu'il fournit une méthode précise pour se déplacer entre la représentation de haut niveau basée sur les particules et le modèle atomique.

L'idée principale de ce chapitre est de montrer comment le modèle tripeptidique, l'algorithme RRT et l'analyse en mode normal peuvent créer un outil efficace pour étudier les transitions conformationnelles, lorsqu'ils sont combinés ensemble. Les expériences réalisées ont montré que le calcul des modes normaux d'une protéine en utilisant les particules au lieu des atomes C_α ne conduit pas à une dégradation de la capacité de prédire les directions du mouvement. Les résultats ont également montré que la méthode proposée est capable de calculer des chemins pour les transitions conformationnelles de différentes longueurs dans des protéines de différentes tailles et de différentes topologies. La performance de notre méthode varie linéairement en fonction du nombre de résidus. En utilisant un seul processeur AMD Opteron 148 à 2,6 GHz , l'étude des transitions prend quelques heures dans de petites protéines et quelques jours dans de grandes protéines, en fonction de la longueur du trajet calculé. Notez cependant que les temps de calcul ont été montrés seulement pour une première implémentation non-optimisée de la méthode. L'amélioration des fonctions les plus coûteuses, telles que la recherche du plus proche voisin, pourrait considérablement accélérer les calculs. L'analyse de la transition conformationnelle de la protéine ADK par notre méthode montre également qu'elle est capable de produire des chemins qui sont compatibles avec les résultats obtenus précédemment.

Conclusion

Nous avons présenté dans cette thèse une approche, appelée modèle tripeptidique, inspirée par la robotique, pour la modélisation des protéines, et nous avons montré comment elle peut être utilisée pour améliorer les simulations moléculaires. Notre approche de modélisation fournit une représentation de haut niveau (gros grains), basée sur une subdivision mécanistique de la protéine sous forme de chaînes cinématiques courtes, appelées tripeptides. Elle fournit également une méthode précise pour générer une représentation détaillée de bas niveau (plein atome) à l'aide de la cinématique inverse 6R en cas de besoin. L'avantage de ce type de modélisation, comme illustré dans cette thèse, est qu'il propose une méthode unifiée pour la mise en œuvre d'une variété d'algorithmes de simulation qui permettent de traiter efficacement les protéines en utilisant la représentation gros grains, mais sans perte de détail au niveau atomique.

En plus de la présentation du modèle tripeptidique, nous avons montré deux applica-

tions de son utilisation pour l'amélioration de simulations moléculaires. Dans la première application, nous avons utilisé le modèle tripeptidique pour la mise en œuvre de nouvelles classes de mouvements pour l'échantillonnage avec Monte Carlo, ainsi que plusieurs autres classes proposées dans ces dernières décennies, et ce afin d'améliorer l'échantillonnage du squelette de la protéine. La flexibilité du modèle tripeptidique nous a également permis de combiner facilement ces classes de mouvement dans une stratégie d'échantillonnage mixte alternant l'utilisation de ces différentes classes. Les simulations effectuées avec deux protéines de différentes tailles et différentes topologies ont validé l'applicabilité de cette approche. Les simulations effectuées ont également montré que la stratégie d'échantillonnage mixte fournit un gain de performance évident sur les autres classes de mouvement. La stratégie mixte a été en mesure d'explorer des conformations qui sont meilleures que les conformations explorées par les autres classes de mouvement, à la fois en termes d'énergie et de variabilité structurelle, sans exiger d'importantes ressources de calcul.

Dans la deuxième application, nous avons présenté une méthode de planification de mouvement pour l'étude des mouvements de grande amplitude dans les protéines. Cette méthode combine le modèle tripeptidique, RRT et l'analyse en mode normal, pour explorer efficacement les transitions conformationnelles dans des protéines ayant plus de mille degrés de liberté. Bien que RRT soit connu pour explorer rapidement des espaces de grande dimension, il peut passer un temps considérable à explorer des parties de l'espace qui ne sont pas directement pertinentes pour la transition conformationnelle étudiée. Cependant, les informations fournies par les modes normaux nous ont permis de surmonter cette limitation en biaisant l'exploration de RRT vers les parties les plus pertinentes de l'espace. En outre, l'utilisation du modèle tripeptidique a réduit le nombre de modes normaux calculés, ce qui a amélioré la performance globale de l'algorithme. Ce modèle a également fourni une méthode précise pour garder une trace des détails au niveau atomique lors de l'exploration. Les simulations réalisées à l'aide de notre méthode ont montré que les réseaux élastiques construits en utilisant le modèle tripeptidique peuvent permettre de prédire la direction du mouvement avec une précision comparable à celle des directions calculées en utilisant les réseaux élastiques C_α. Les simulations ont aussi montré que la méthode introduite peut calculer des chemins de transition entre des conformations de protéines de différentes tailles et de différentes topologies, et que ses performances varient linéairement en fonction du nombre de résidus. Une analyse détaillée de la trajectoire calculée pour la protéine ADK a également montré que la méthode produit des chemins qui sont en accord avec les résultats obtenus avec d'autres méthodes plus coûteuses.

Travaux futurs

Le travail présenté sur l'application du modèle tripeptidique pour améliorer les simulations de Monte Carlo peut être étendu et approfondi. Tout d'abord, les nouvelles classes de mouvement, qui sont basées sur la perturbation d'une particule ou d'un groupe de particules, doivent être testées et comparées plus en détail avec les autres classes de mouvement disponibles. Les questions qui demandent encore des réponses plus précises concernent le type de topologie des protéines et les problèmes de simulation moléculaire qui sont le plus adaptés à ces classes de mouvement, et quelles variantes de ces classes de mouvement offrent les meilleures performances. De même, des tests supplémentaires doivent être effectués afin d'identifier des combinaisons optimales de classes de mouvement ou les probabilités d'utilisation dans la stratégie mixte pour les différentes topologies de protéines et les problèmes de simulation moléculaire.

Le RRT guidé par les modes normaux a été utilisé dans cette thèse pour trouver un chemin de transition entre deux conformations connues. Toutefois, les applications de cette méthode dépassent largement ce contexte, car elle peut être utilisée pour étudier d'autres types de problèmes qui nécessitent une exploration de l'espace conformationnel. Par exemple, notre méthode peut être utilisée pour prédire les conformations les plus probables que la protéine peut atteindre à partir d'une conformation donnée. D'autres axes de recherche possibles ont été mis en évidence également dans la conclusion du chapitre 4, telles que l'utilisation des RRT parallélisés [Devaurs 11], du RRT bi-directionnel [Kuffner Jr 00] et du RRT avec Transitions [Jaillet 10, Jaillet 11]. Ces variantes de RRT ont permis d'améliorer les performances de RRT, ce qui fait d'elles des candidates pour l'amélioration des performances de notre méthode. Une autre direction de recherche est l'exploration de voies possibles pour le traitement de la flexibilité des chaînes latérales, pour remplacer l'étape de minimisation actuellement effectuée dans notre méthode.

Notre objectif à court terme est l'introduction de plus d'abstraction dans le modèle tripeptidique. Actuellement, le modèle ne comporte que deux couches : la procédure simplifiée de représentation à base de particules et la représentation atomique. Ceci peut être une limitation, en particulier lorsque de très grosses protéines sont traitées. Il serait possible, par exemple, d'ajouter une autre couche reliant plusieurs tripeptides consécutifs en un fragment. Cela peut être utile pour le traitement de grosses protéines avec une flexibilité limitée. Il est également possible de subdiviser la protéine en fragments de longueurs variables en fonction de la souplesse des différentes parties de la protéine. Les structures secondaires, par exemple, sont connues pour être moins souples que les boucles des protéines, ce qui permet d'affirmer qu'il serait raisonnable d'utiliser des fragments comprenant plus de trois résidus à ce niveau là. Un autre objectif à court terme est d'utiliser (ou de développer) un modèle énergétique plus approprié pour le modèle tripep-

tidique. Le choix du modèle énergétique est connu pour avoir un effet important sur les résultats obtenus, ce qui explique pourquoi il est indispensable d'étudier plus avant dans cette direction.

Nous prévoyons également d'évaluer l'utilisation du modèle tripeptidique pour l'étude des interactions moléculaires, où la flexibilité totale des protéines en interaction pose un véritable défi. En utilisant notre modèle tripeptidique, la flexibilité des protéines peut être traitée grâce aux particules, au lieu d'avoir à faire face à tous les degrés de liberté. De plus, les modèles atomiques des protéines peuvent être générés chaque fois que cela est nécessaire. D'autre part, les interactions moléculaires induisent des déformations qui sont différentes de celles qui se produisent naturellement, en réponse à des forces internes. Une façon d'étudier les déformations induites par les interactions est d'utiliser les modes statiques [Brut 09], qui fournissent des directions de mouvement se produisant en réponse à l'application de forces extérieures. Par conséquent, les modes statiques pourraient remplacer les modes normaux dans la méthode exposée dans le chapitre 4, afin de guider l'exploration vers de potentielles conformations induites par les interactions.

Bibliography

[Abagyan 94] R. Abagyan, M. Totrov & D. Kuznetsov. *ICM - A new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation.* Journal of Computational Chemistry, vol. 15, no. 5, pages 488–506, 1994.

[Agarwal 04] P. Agarwal, L. Guibas, A. Nguyen, D. Russel & L. Zhang. *Collision detection for deforming necklaces.* Computational Geometry, vol. 28, no. 2-3, pages 137–163, 2004.

[Alexandrov 05] V. Alexandrov, U. Lehnert, N. Echols, D. Milburn, D. Engelman & M. Gerstein. *Normal modes for predicting protein motions: a comprehensive database assessment and associated Web tool.* Protein science, vol. 14, no. 3, pages 633–643, 2005.

[Altis 07] A. Altis, P.H. Nguyen, R. Hegger & G. Stock. *Dihedral angle principal component analysis of molecular dynamics simulations.* The Journal of Chemical Physics, vol. 126, no. 24, page 244111, 2007.

[Amato 98] N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones & D. Vallejo. *OBPRM: An obstacle-based PRM for 3D workspaces.* In Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics, pages 155–168, 1998.

[Amato 03] N.M. Amato, K.A. Dill & G. Song. *Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures.* Journal of Computational Biology, vol. 10, no. 3-4, pages 239–255, 2003.

[Anderson 99] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney & D. Sorensen. *LAPACK Users' Guide.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 3rd edition edition, 1999.

[Angeles 07] J. Angeles. Fundamentals of robotic mechanical systems: Theory, methods, and algorithms. Mechanical Engineering Series. Springer, 2007.

[Apaydin 01] M.S. Apaydin, A.P. Singh, D.L. Brutlag & J.C. Latombe. *Capturing molecular energy landscapes with probabilistic conformational*

roadmaps. In Proceedings of the IEEE International Conference on Robotics and Automation, volume 1, pages 932–939, 2001.

[Apaydin 02] M.S. Apaydin, C.E. Guestrin, C. Varma, D.L. Brutlag & J.C. Latombe. *Stochastic roadmap simulation for the study of ligand-protein interactions*. Bioinformatics, vol. 18, no. Suppl 2, pages S18–S26, 2002.

[Apaydin 03] M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, J.C. Latombe & C. Varma. *Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion*. Journal of Computational Biology, vol. 10, no. 3-4, pages 257–281, 2003.

[Apaydin 04] M.S. Apaydin, D.L. Brutlag, D. Hsu & J.C. Latombe. *Stochastic conformational roadmaps for computing ensemble properties of molecular motion*. Algorithmic Foundations of Robotics V, pages 131–147, 2004.

[Apostolakis 98] J. Apostolakis, A. Plückthun & A. Caflisch. *Docking small ligands in flexible binding sites*. Journal of Computational Chemistry, vol. 19, no. 1, pages 21–37, 1998.

[Atilgan 01] AR Atilgan, SR Durell, RL Jernigan, MC Demirel, O. Keskin & I. Bahar. *Anisotropy of fluctuation dynamics of proteins with an elastic network model*. Biophysical Journal, vol. 80, no. 1, pages 505–515, 2001.

[Atramentov 02] A. Atramentov & S. M. LaValle. *Efficient Nearest Neighbor Searching for Motion Planning*. Proceedings of the IEEE International Conference on Robotics and Automation, pages 632–637, 2002.

[Bahar 97] I. Bahar, A. R. Atilgan & B. Erman. *Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential*. Folding and Design, vol. 2, no. 3, pages 173–181, 1997.

[Balbach 95] J. Balbach, V. Forge, N.A.J. van Nuland, S.L. Winder, P.J. Hore & C.M. Dobson. *Following protein folding in real time using NMR spectroscopy*. Nature Structural & Molecular Biology, vol. 2, no. 10, pages 865–870, 1995.

[Banaszak 00] L. Banaszak. Foundations of structural biology. Academic Press, 2000.

[Barbe 11] S. Barbe, J. Cortés, T. Siméon, P. Monsan, M. Remaud-Siméon & I. André. *A mixed molecular modelling - robotics approach to investigate lipase large molecular motions*. Proteins: Structure, Function and Bioinformatics, vol. 79, no. 8, pages 2517–2529, 2011.

[Berman 02] H.M. Berman, T. Battistuz, TN Bhat, W.F. Bluhm, P.E. Bourne, K. Burkhardt, Z. Feng, G.L. Gilliland, L. Iype, S. Jain *et al. The protein data bank.* Acta Crystallographica Section D: Biological Crystallography, vol. 58, no. 6, pages 899–907, 2002.

[Bernadu 05] P. Bernadu, L. Blanchard, P. Timmins, D. Marion, R.W.H. Ruigrok & M. Blackledge. *A structural model for unfolded proteins from residual dipolar couplings and small-angle x-ray scattering.* Proceedings of the National Academy of Sciences of the U.S.A., vol. 102, pages 17002–17005, 2005.

[Betancourt 05] M. R. Betancourt. *Efficient Monte Carlo Trial Moves for Polypeptide Simulations.* Journal of Chemical Physics, vol. 123, page 174905, 2005.

[Bondi 64] A. Bondi. *Van der Waals Volumes and Radii.* Journal of Physical Chemistry, vol. 68, pages 441–451, 1964.

[Bonneau 01] Richard Bonneau & David Baker. *Ab Initio protein structure prediction: progress and prospects.* Annual Review of Biophysics and Biomolecular Structure, vol. 30, no. 1, pages 173–189, 2001.

[Brooks 85] B. Brooks & M. Karplus. *Normal modes for specific motions of macromolecules: application to the hinge-bending mode of lysozyme.* Proceedings of the National Academy of Sciences, vol. 82, no. 15, pages 4995–4999, 1985.

[Bruce 02] J. Bruce & M. Veloso. *Real-time randomized path planning for robot navigation.* In IEEE/RSJ International Conference on Intelligent Robots and Systems, volume 3, pages 2383–2388, 2002.

[Brunette 08] TJ Brunette & O. Brock. *Guiding conformation space search with an all-atom energy potential.* Proteins: Structure, Function, and Bioinformatics, vol. 73, no. 4, pages 958–972, 2008.

[Brut 09] M. Brut, A. Estève, G. Landa, G. Renvez & M. Djafari Rouhani. *The Static Modes: An alternative approach for the treatment of macro-and bio-molecular induced-fit flexibility.* The European Physical Journal E: Soft Matter and Biological Physics, vol. 28, no. 1, pages 17–25, 2009.

[Bryngelson 95] J.D. Bryngelson, J.N. Onuchic, N.D. Socci & P.G. Wolynes. *Funnels, pathways, and the energy landscape of protein folding: a synthesis.* Proteins: Structure, Function, and Bioinformatics, vol. 21, no. 3, pages 167–195, 1995.

[Burkert 82] U. Burkert & N.L. Allinger. Molecular mechanics. American Chemical Society, 1982.

[Canny 88] John F. Canny. The complexity of robot motion planning. MIT Press, Cambridge, MA, USA, 1988.

[Canutescu 03] A.A. Canutescu & R.L. Dunbrack Jr. *Cyclic coordinate descent: A robotics algorithm for protein loop closure.* Protein Science, vol. 12, no. 5, pages 963–972, 2003.

[Case 06] D.A. Case, TA Darden, T.E. Cheatham III, CL Simmerling, J. Wang, R.E. Duke, R. Luo, K.M. Merz, D.A. Pearlman, M. Crowley *et al. AMBER 9.* University of California, San Francisco, 2006.

[Cavanagh 06] John Cavanagh. Protein NMR spectroscopy: principles and practices. Royal Society of Chemistry, 2006.

[Cavasotto 05a] C.N. Cavasotto, J.A. Kovacs & R.A. Abagyan. *Representing receptor flexibility in ligand docking through relevant normal modes.* Journal of the American Chemical Society, vol. 127, no. 26, pages 9632–9640, 2005.

[Cavasotto 05b] C.N. Cavasotto, A.J.W. Orry & R.A. Abagyan. *The challenge of considering receptor flexibility in ligand docking and virtual screening.* Current Computer-Aided Drug Design, vol. 1, no. 4, pages 423–440, 2005.

[Chan 97] C.K. Chan, Y. Hu, S. Takahashi, D.L. Rousseau, W.A. Eaton & J. Hofrichter. *Submillisecond protein folding kinetics studied by ultrarapid mixing.* Proceedings of the National Academy of Sciences of the United States of America, vol. 94, no. 5, pages 1779–1784, 1997.

[Cheng 02] P. Cheng & S.M. LaValle. *Resolution complete rapidly-exploring random trees.* In Proceedings of the IEEE International Conference on Robotics and Automation, volume 1, pages 267–272, 2002.

[Chiang 06] T.H. Chiang, M. Apaydin, D. Brutlag, D. Hsu & J.C. Latombe. *Predicting experimental quantities in protein folding kinetics using stochastic roadmap simulation.* In Research in Computational Molecular Biology, pages 410–424. Springer, 2006.

[Chiang 07] T.H. Chiang, M.S. Apaydin, D.L. Brutlag, D. Hsu & J.C. Latombe. *Using stochastic roadmap simulation to predict experimental quantities in protein folding kinetics: folding rates and phi-values.* Journal of Computational Biology, vol. 14, no. 5, pages 578–593, 2007.

[Choset 05] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki & S. Thrun. Principles of robot motion: theory, algorithms, and implementation. Intelligent robotics and autonomous agents. MIT Press, 2005.

[Cohen 95] J.D. Cohen, M.C. Lin, D. Manocha & M. Ponamgi. *I-COLLIDE: An interactive and exact collision detection system for large-scale environments.* In Proceedings of the 1995 symposium on Interactive 3D graphics, pages 189–196. ACM, 1995.

[Cortés 04] J. Cortés, T. Siméon, M. Remaud-Siméon & V. Tran. *Geometric algorithms for the conformational analysis of long protein loops.* Journal of Computational Chemistry, vol. 25, no. 7, pages 956–967, 2004.

[Cortés 05a] J. Cortés & T. Siméon. *Sampling-based motion planning under kinematic loop-closure constraints.* Algorithmic Foundations of Robotics VI, pages 75–90, 2005.

[Cortés 05b] J. Cortés, T. Siméon, V. Ruiz de Angulo, D. Guieysse, M. Remaud-Siméon & V. Tran. *A path planning approach for computing large-amplitude motions of flexible molecules.* Bioinformatics, vol. 21, no. suppl 1, pages i116–i125, 2005.

[Cortés 07] J. Cortés, L. Jaillet & T. Siméon. *Molecular disassembly with RRT-like algorithms.* In Proceedings of the IEEE International Conference on Robotics and Automation, pages 3301–3306, 2007.

[Cortés 08] J. Cortés, L. Jaillet & T. Siméon. *Disassembly path planning for complex articulated objects.* IEEE Transactions on Robotics, vol. 24, no. 2, pages 475–481, 2008.

[Cortés 10a] J. Cortés, S. Carrión, D. Curcó, M. Renaud & C. Alemán. *Relaxation of amorphous multichain polymer systems using inverse kinematics.* Polymer, vol. 51, no. 17, pages 4008–4014, 2010.

[Cortés 10b] J. Cortés, D.T. Le, R. Iehl & T. Siméon. *Simulating ligand-induced conformational changes in proteins using a mechanical disassembly method.* Physical Chemistry Chemical Physics, vol. 12, no. 29, pages 8268–8276, 2010.

[Coutsias 04] E.A. Coutsias, C. Seok, M.P. Jacobson & K.A. Dill. *A kinematic view of loop closure.* Journal of computational chemistry, vol. 25, no. 4, pages 510–528, 2004.

[Craig 89] J.J. Craig. Introduction to robotics: Mechanics and control. Addison-Wesley, 1989.

[Cui 06] Q. Cui & I. Bahar. Normal mode analysis: theory and applications to biological and chemical systems. Chapman and Hall/CRC mathematical and computational biology series. Chapman & Hall/CRC, 2006.

[Das 06] P. Das, M. Moll, H. Stamati, L.E. Kavraki & C. Clementi. *Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction.* Proceedings of the National Academy of Sciences, vol. 103, no. 26, pages 9885–9890, 2006.

[de Angulo 05] V.R. de Angulo, J. Cortés & T. Siméon. *BioCD: An efficient algorithm for self-collision and distance computation between highly*

articulated molecular models. In Robotics: Science And Systems I, pages 241–248. MIT Press, 2005.

[Derreumaux 99] P. Derreumaux. *From polypeptide sequences to structures using Monte Carlo simulations and an optimized potential.* Journal of Chemical Physics, vol. 111, no. 5, pages 2301–2310, 1999.

[Devaurs 11] D. Devaurs, T. Siméon & J. Cortés. *Parallelizing RRT on distributed-memory architectures.* In Proceedings of the IEEE International Conference on Robotics and Automation, pages 2261–2266. IEEE, 2011.

[Dijkstra 59] E.W. Dijkstra. *A note on two problems in connexion with graphs.* Numerische Mathematik, vol. 1, no. 1, pages 269–271, 1959.

[Dill 08] K.A. Dill, S.B. Ozkan, M.S. Shell & T.R. Weikl. *The protein folding problem.* Annual review of biophysics, vol. 37, pages 289–316, 2008.

[Dobson 03] C.M. Dobson. *Protein folding and misfolding.* Nature, vol. 426, no. 6968, pages 884–890, 2003.

[Dodd 93] L. R. Dodd, T. D. Boone & D. N. Theodorou. *A Concerted Rotation Algorithm for Atomistic Monte Carlo Simulation of Polymer Melts and Glasses.* Molecular Physics, vol. 78, no. 4, pages 961–996, 1993.

[Dyson 04] H.J. Dyson & P.E. Wright. *Unfolded proteins and protein folding studied by NMR.* Chem. Rev, vol. 104, no. 8, pages 3607–3622, 2004.

[Enosh 08] A. Enosh, B. Raveh, O. Furman-Schueler, D. Halperin & N. Ben-Tal. *Generation, comparison, and merging of pathways between protein conformations: Gating in K-channels.* Biophysical journal, vol. 95, no. 8, pages 3850–3860, 2008.

[Eyal 06] E. Eyal, L.W. Yang & I. Bahar. *Anisotropic network model: systematic evaluation and a new web interface.* Bioinformatics, vol. 22, no. 21, pages 2619–2627, 2006.

[Falicov 96] A. Falicov & F.E. Cohen. *A surface of minimum area metric for the structural comparison of proteins.* Journal of molecular biology, vol. 258, no. 5, pages 871–892, 1996.

[Feng 09] Y. Feng, L. Yang, A. Kloczkowski & R.L. Jernigan. *The energy profiles of atomic conformational transition intermediates of adenylate kinase.* Proteins: Structure, Function, and Bioinformatics, vol. 77, no. 3, pages 551–558, 2009.

[Fodor 02] I. K. Fodor. *A survey of dimension reduction techniques.* Rapport technique UCRL-ID-148494, Lawrence Livermore National Lab, June 2002.

[Frenkel 02] D. Frenkel & B. Smit. Understanding Molecular Simulations: From Algorihtms to Applications. Academic Press, 2002.

[Geraerts 04] R. Geraerts & M. Overmars. *A comparative study of probabilistic roadmap planners.* Algorithmic Foundations of Robotics V, pages 43–58, 2004.

[Gipson 12] B. Gipson, D. Hsu, L. E. Kavraki & Latombe J.-C. *Computational Models of Protein Kinematics and Dynamics: Beyond Simulation.* Annual Review of Analytical Chemistry, vol. 5, pages 273–291, 2012.

[Gō 70] N. Gō & H.A. Scheraga. *Ring Closure and Local Conformational Deformations of Chain Molecules.* Macromolecules, vol. 3, pages 178–187, 1970.

[Gō 90] N. Gō. *A theorem on amplitudes of thermal atomic fluctuations in large molecules assuming specific conformations calculated by normal mode analysis.* Biophysical Chemistry, vol. 35, no. 1, pages 105–112, 1990.

[Goldberg 89] D.E. Goldberg. Genetic algorithms in search, optimization, and machine learning. Addison-wesley, 1989.

[Goldberg 95] Ken Goldberg. *Completeness in robot motion planning.* In Proceedings of the workshop on Algorithmic foundations of robotics, pages 419–429, Natick, MA, USA, 1995. A. K. Peters, Ltd.

[Golub 96] G. H. Golub & C. F. Van Loan. Matrix computations. Johns Hopkins University Press, 3rd edition edition, 1996.

[Goodsell 96] D.S. Goodsell, G.M. Morris & A.J. Olson. *Automated docking of flexible ligands: applications of AutoDock.* Journal of Molecular Recognition, vol. 9, no. 1, pages 1–5, 1996.

[Gottschalk 96] S. Gottschalk, M. C. Lin & D. Manocha. *OBBTree: a hierarchical structure for rapid interference detection.* In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 171–180. ACM, 1996.

[Griffiths 05] D.J. Griffiths. Introduction to quantum mechanics. Pearson Prentice Hall, 2005.

[Guieysse 08] D. Guieysse, J. Cortés, S. Puech-Guenot, S. Barbe, V. Lafaquière, P. Monsan, T. Siméon, I. André & M. Remaud-Siméon. *A Structure-Controlled Investigation of Lipase Enantioselectivity by a Path-Planning Approach.* ChemBioChem, vol. 9, no. 8, pages 1308–1317, 2008.

[Hajduk 07] P.J. Hajduk & J. Greer. *A decade of fragment-based drug design: strategic advances and lessons learned.* Nature Reviews Drug Discovery, vol. 6, no. 3, pages 211–219, 2007.

[Hart 72] P.E. Hart, N.J. Nilsson & B. Raphael. *Correction to "A formal basis for the heuristic determination of minimum cost paths"*. ACM SIGART Bulletin, no. 37, pages 28–29, 1972.

[Haspel 10] N. Haspel, M. Moll, M. Baker, W. Chiu & L. Kavraki. *Tracing conformational changes in proteins*. BMC Structural Biology, vol. 10, no. Suppl 1, page S1, 2010.

[Hinsen 98] K. Hinsen. *Analysis of domain motions by approximate normal mode calculations*. Proteins: Structure, Function, and Bioinformatics, vol. 33, no. 3, pages 417–429, 1998.

[Hinsen 99] K. Hinsen, A. Thomas & M.J. Field. *Analysis of domain motions in large proteins*. Proteins: Structure, Function, and Bioinformatics, vol. 34, no. 3, pages 369–382, 1999.

[Holm 93] L. Holm, C. Sander *et al. Protein structure comparison by alignment of distance matrices*. Journal of molecular biology, vol. 233, no. 1, pages 123–138, 1993.

[Hsu 97] D. Hsu, J.C. Latombe & R. Motwani. *Path planning in expansive configuration spaces*. In Proceedings of the IEEE International Conference on Robotics and Automation, volume 3, pages 2719–2726, 1997.

[Iehl 12] R. Iehl, J. Cortés & T. Siméon. *Costmap planning in high dimensional configuration spaces*. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2012. In press.

[Jagodzinski 07] F. Jagodzinski & O. Brock. *Towards a mechanistic view of protein motion*. In 46th IEEE Conference on Decision and Control, pages 4557–4562. IEEE, 2007.

[Jaillet 10] L. Jaillet, J. Cortés & T. Siméon. *Sampling-based path planning on configuration-space costmaps*. IEEE Transactions on Robotics, vol. 26, no. 4, pages 635–646, 2010.

[Jaillet 11] L. Jaillet, F.J. Corcho, J.J. Pérez & J. Cortés. *Randomized tree construction algorithm to explore energy landscapes*. Journal of Computational Chemistry, vol. 32, no. 16, pages 3464–3474, 2011.

[Jeong 06] J.I. Jeong, E.E. Lattman & G.S. Chirikjian. *A method for finding candidate conformations for molecular replacement using relative rotation between domains of a known structure*. Acta Crystallographica Section D: Biological Crystallography, vol. 62, no. 4, pages 398–409, 2006.

[Jiménez 01] P. Jiménez, F. Thomas & C. Torras. *3D collision detection: a survey*. Computers & Graphics, vol. 25, no. 2, pages 269–285, 2001.

[Jolliffe 02] IT Jolliffe. Principal component analysis. Springer Verlag, 2002.

[Jones 93] C.M. Jones, E.R. Henry, Y. Hu, C.K. Chan, S.D. Luck, A. Bhuyan, H. Roder, J. Hofrichter & W.A. Eaton. *Fast events in protein folding initiated by nanosecond laser photolysis.* Proceedings of the National Academy of Sciences of the United States of America, vol. 90, no. 24, pages 11860–11864, 1993.

[Jones 97a] G. Jones, P. Willett, R.C. Glen, A.R. Leach & R. Taylor. *Development and validation of a genetic algorithm for flexible docking.* Journal of Molecular Biology, vol. 267, no. 3, pages 727–748, 1997.

[Jones 97b] G. Jones, P. Willett, R.C. Glen, A.R. Leach & R. Taylor. *Development and validation of a genetic algorithm for flexible docking.* Journal of Molecular Biology, vol. 267, no. 3, pages 727–748, 1997.

[Jorgensen 96] W. L. Jorgensen, D. S. Maxwell & J. Tirado-Rives. *Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids.* Journal of the American Chemical Society, vol. 118, no. 45, pages 11225–11236, 1996.

[Kavraki 96] L.E. Kavraki, P. Svestka, J.C. Latombe & M.H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces.* IEEE transactions on Robotics and Automation, vol. 12, no. 4, pages 566–580, 1996.

[Kavraki 07] L.E. Kavraki. *Geometric Methods in Structural Computational Biology.* 2007.

[Kim 02] M.K. Kim, R.L. Jernigan & G.S. Chirikjian. *Efficient generation of feasible pathways for protein conformational transitions.* Biophysical Journal, vol. 83, no. 3, pages 1620–1630, 2002.

[Kirillova 08] S. Kirillova, J. Cortés, A. Stefaniu & T. Siméon. *An NMA-guided path planning approach for computing large-amplitude conformational changes in proteins.* Proteins: Structure, Function, and Bioinformatics, vol. 70, no. 1, pages 131–143, 2008.

[Kirkpatrick 83] S. Kirkpatrick, C.D. Gelatt & M.P. Vecchi. *Optimization by simulated annealing.* Science, vol. 220, no. 4598, pages 671–680, 1983.

[Koliński 10] A. Koliński. Multiscale approaches to protein modeling. Springer Verlag, 2010.

[Kolodny 05] R. Kolodny, L. Guibas, M. Levitt & P. Koehl. *Inverse kinematics in biology: The protein loop closure problem.* International Journal of Robotics Research, vol. 24, no. 2-3, pages 151–163, 2005.

[Krivov 09] G. G. Krivov, M. V. Shapovalov & R. L. Dunbrack Jr. *Improved Prediction of Protein Side-chain Conformations with SCWRL4.* Proteins: Structure, Function, and Bioinformatics, vol. 77, no. 4, pages 778–795, 2009.

[Kuffner Jr 00] J.J. Kuffner Jr & S.M. LaValle. *RRT-connect: An efficient approach to single-query path planning.* In Proceedings of the IEEE International Conference on Robotics and Automation, volume 2, pages 995–1001, 2000.

[Ladd 05] A. M. Ladd & L. E. Kavraki. Fast tree-based exploration of state space for robots with dynamics, pages 297–312. Springer, 2005.

[Lafaquière 09] V. Lafaquière, S. Barbe, S. Puech-Guenot, D. Guieysse, J. Cortés, P. Monsan, T. Siméon, I. André & M. Remaud-Siméon. *Control of lipase enantioselectivity by engineering the substrate binding site and access channel.* ChemBioChem, vol. 10, no. 17, pages 2760–2771, 2009.

[Laio 02] A. Laio & M. Parrinello. *Escaping free-energy minima.* Proceedings of the National Academy of Sciences of the United States of America, vol. 99, no. 20, pages 12562–12566, 2002.

[Lal 69] M. Lal. *ÔMonte CarloÕcomputer simulation of chain molecules. I.* Molecular physics, vol. 17, no. 1, pages 57–64, 1969.

[Landau 05] D.P. Landau & K. Binder. A guide to monte carlo simulations in statistical physics. Cambridge University Press, 2005.

[Lang 09] P.T. Lang, S.R. Brozell, S. Mukherjee, E.F. Pettersen, E.C. Meng, V. Thomas, R.C. Rizzo, D.A. Case, T.L. James & I.D. Kuntz. *DOCK 6: Combining techniques to model RNA–small molecule complexes.* RNA, vol. 15, no. 6, pages 1219–1230, 2009.

[Latombe 90] J.C. Latombe. Robot motion planning. Springer Verlag, 1990.

[LaValle 00] S.M. LaValle, P.W. Finn, L.E. Kavraki & J.C. Latombe. *A randomized kinematics-based approach to pharmacophore-constrained conformational search and database screening.* Journal of Computational Chemistry, vol. 21, no. 9, pages 731–747, 2000.

[LaValle 01a] S. LaValle & J. Kuffner. *Rapidly-exploring random trees: Progress and prospects.* In Algorithmic and computational robotics: new directions: the fourth Workshop on the Algorithmic Foundations of Robotics, pages 293–308, 2001.

[LaValle 01b] S.M. LaValle & J.J. Kuffner Jr. *Randomized kinodynamic planning.* The International Journal of Robotics Research, vol. 20, no. 5, pages 378–400, 2001.

[LaValle 06] S.M. LaValle. Planning algorithms. Cambridge University Press, 2006.

[Leach 01] A. R. Leach. Molecular modelling: Principles and applications. Pearson Education, 2001.

[Lee 88a] H. Y. Lee & C. G. Liang. *Displacement Analysis of the General Spatial 7-Link 7R Mechanisms.* Mechanism and Machine Theory, vol. 23, no. 3, pages 219–226, 1988.

[Lee 88b] H. Y. Lee & C. G. Liang. *A New Vector Theory for the Analysis of Spatial Mechanisms.* Mechanism and Machine Theory, vol. 23, no. 3, pages 209–217, 1988.

[Lee 05] A. Lee, I. Streinu & O. Brock. *A methodology for efficiently sampling the conformation space of molecular structures.* Physical biology, vol. 2, pages S108–S115, 2005.

[Lengauer 96] T. Lengauer & M. Rarey. *Computational methods for biomolecular docking.* Current Opinion in Structural Biology, vol. 6, no. 3, pages 402–406, 1996.

[Leontidis 94] E. Leontidis, J. J. de Pablo, M. Laso & U. W. Suter. *A Critical Evaluation of Novel Algorithms for the Off-Lattice Monte Carlo Simulation of Condensed Polymer Phases.* Advances in Polymer Science, vol. 116, pages 283–318, 1994.

[Li 08] D. Li, H. Yang, L. Han & S. Huo. *Predicting the Folding Pathway of Engrailed Homeodomain with a Probabilistic Roadmap Enhanced Reaction-Path Algorithm.* Biophysical journal, vol. 94, no. 5, pages 1622–1629, 2008.

[Lin 03] M. Lin & D. Manocha. *Collision and Proximity Queries.* In Handbook of Discrete and Computational Geometry. 2003.

[Lindemann 05] S.R. Lindemann & S.M. LaValle. *Current issues in sampling-based motion planning.* Robotics Research, pages 36–54, 2005.

[Lotan 02] I. Lotan, F. Schwarzer, D. Halperin & J.C. Latombe. *Efficient maintenance and self-collision testing for kinematic chains.* In Proceedings of the eighteenth annual symposium on Computational geometry, pages 43–52. ACM, 2002.

[Lotan 04] I. Lotan & F. Schwarzer. *Approximation of protein structure for fast similarity measures.* Journal of Computational Biology, vol. 11, no. 2-3, pages 299–317, 2004.

[Lozano-Peréz 83] T. Lozano-Peréz. *Spatial planning: A configuration space approach.* IEEE Transactions on Computers, vol. 32, no. 2, pages 108–120, 1983.

[Mackerell Jr 04] A.D. Mackerell Jr. *Empirical force fields for biological macromolecules: overview and issues.* Journal of Computational Chemistry, vol. 25, no. 13, pages 1584–1604, 2004.

[Manocha 94] D. Manocha & J.F. Canny. *Efficient Inverse Kinematics of General* $6R$ *Manipulators.* IEEE Transactions on Robotics and Automation, vol. 10(5), pages 648–657, 1994.

[Manocha 95] D. Manocha, Y. Zhu & W. Wright. *Conformational analysis of molecular chains using nano-kinematics.* Computer Applications in the Biosciences: CABIOS, vol. 11, no. 1, pages 71–86, 1995.

[Maragakis 05] P. Maragakis & M. Karplus. *Large amplitude conformational change in proteins explored with a plastic network model: adenylate kinase.* Journal of Molecular Biology, vol. 352, pages 807–822, 2005.

[Marinari 92] E. Marinari & G. Parisi. *Simulated tempering: a new Monte Carlo scheme.* Europhysics letters, vol. 19, no. 6, pages 451–458, 1992.

[Marques 95] O. Marques & Y.H. Sanejouand. *Hinge-bending motion in citrate synthase arising from normal mode calculations.* Proteins: Structure, Function, and Bioinformatics, vol. 23, no. 4, pages 557–560, 1995.

[Metropolis 53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller & E.Teller. *Equation of State Calculations by Fast Computing Machines.* Journal of Chemical Physics, vol. 21, pages 1087–1092, 1953.

[Miyashita 03] O. Miyashita, JN Onuchic & PG Wolynes. *Nonlinear elasticity, proteinquakes, and the energy landscapes of functional transitions in proteins.* Proceedings of the National Academy of Sciences of the United States of America, vol. 100, no. 22, page 12570, 2003.

[Moll 07] M. Moll, D. Schwarz & Lydia E. Kavraki. Roadmap methods for protein folding. Humana Press, 2007.

[Monticelli 08] L. Monticelli, S.K. Kandasamy, X. Periole, R.G. Larson, D.P. Tieleman & S.J. Marrink. *The MARTINI coarse-grained force field: extension to proteins.* Journal of Chemical Theory and Computation, vol. 4, no. 5, pages 819–834, 2008.

[Mouawad 96] L. Mouawad & D. Perahia. *Motions in hemoglobin studied by normal mode analysis and energy minimization: evidence for the existence of tertiary T-like, quaternary R-like intermediate structures.* Journal of Molecular Biology, vol. 258, no. 2, pages 393–410, 1996.

[Mu 05] Y. Mu, P.H. Nguyen & G. Stock. *Energy landscape of a small peptide revealed by dihedral angle principal component analysis.* Proteins: Structure, Function, and Bioinformatics, vol. 58, no. 1, pages 45–52, 2005.

[Müller 92] C.W. Müller & G.E. Schulz. *Structure of the complex between adenylate kinase from Escherichia coli and the inhibitor Ap5A refined at 1.9 Å resolution* 1:: A model for a catalytic transition state.* Journal of Molecular Biology, vol. 224, no. 1, pages 159–177, 1992.

[Müller 96] CW Müller, GJ Schlauderer, J. Reinstein & GE Schulz. *Adenylate kinase motions during catalysis: an energetic counterweight balancing substrate binding.* Structure, vol. 4, no. 2, pages 147–156, 1996.

[Muñoz 08] V. Muñoz. Protein folding, misfolding and aggregation: classical themes and novel approaches. RSC biomolecular sciences. Royal Society of Chemistry, 2008.

[Noonan 05] K. Noonan, D. O'Brien & J. Snoeyink. *Probik: Protein backbone motion by inverse kinematics.* The International Journal of Robotics Research, vol. 24, no. 11, pages 971–982, 2005.

[Onuchic 04] J.N. Onuchic & P.G. Wolynes. *Theory of protein folding.* Current Opinion in Structural Biology, vol. 14, no. 1, pages 70–75, 2004.

[Pain 00] R.H. Pain. Mechanisms of protein folding. Frontiers in molecular biology. Oxford University Press, 2000.

[Pak 00] Y. Pak & S. Wang. *Application of a molecular dynamics simulation method with a generalized effective potential to the flexible molecular docking problems.* The Journal of Physical Chemistry B, vol. 104, no. 2, pages 354–359, 2000.

[Parsons 94] D. Parsons & J. Canny. *Geometric problems in molecular biology and robotics.* In Proceedings of the International Conference on Intelligent Systems for Molecular Biology, pages 322–220, 1994.

[Plaku 07a] E. Plaku, H. Stamati, C. Clementi & L. E. Kavraki. *Fast and Reliable Analysis of Molecular Motion Using Proximity Relations and Dimensionality Reduction.* Proteins: Structure, Function, and Bioinformatics, vol. 67, no. 4, pages 897–907, 2007.

[Plaku 07b] E. Plaku, H. Stamati, C. Clementi & L.E. Kavraki. *Fast and reliable analysis of molecular motion using proximity relations and dimensionality reduction.* Proteins: Structure, Function, and Bioinformatics, vol. 67, no. 4, pages 897–907, 2007.

[Ponder 03] J.W. Ponder & D.A. Case. *Force fields for protein simulations.* Advances in protein chemistry, vol. 66, pages 27–85, 2003.

[Rangwala 10] H. Rangwala & G. Karypis. Protein structure methods and algorithms, volume 14 of *Wiley Series in Bioinformatics: Computational Techniques and Engineering.* John Wiley & Sons, 2010.

[Rao 73] S.T. Rao & M.G. Rossmann. *Comparison of super-secondary structures in proteins.* Journal of molecular biology, vol. 76, no. 2, pages 241–256, 1973.

[Rapaport 07] D. C. Rapaport. The art of molecular dynamics simulation. Academic Press, 2007.

[Rarey 96] M. Rarey, B. Kramer, T. Lengauer & G. Klebe. *A fast flexible docking method using an incremental construction algorithm.* Journal of Molecular Biology, vol. 261, no. 3, pages 470–489, 1996.

[Raveh 09] B. Raveh, A. Enosh, O. Schueler-Furman & D. Halperin. *Rapid Sampling of Molecular Motions with Prior Information Constraints.* PLoS Computational Biology, vol. 5, no. 2, page e1000295, 2009.

[Reif 79] John H. Reif. *Complexity of the mover's problem and generalizations.* In Proceedings of the 20th Annual Symposium on Foundations of Computer Science, pages 421–427. IEEE Computer Society, 1979.

[Renaud 00] M. Renaud. *A Simplified Inverse Kinematic Model Calculation Method for all 6R type Manipulators.* Proceedings of the International Conference on Mechanical Design and Production, pages 15–25, 2000.

[Renaud 06] M. Renaud. *Calcul des Modéles Géométriques Inverses des Robots Manipulateurs 6R.* Rapport LAAS 06332, LAAS, Toulouse, 2006.

[Robustelli 10] P. Robustelli, K. Kohlhoff, A. Cavalli & M. Vendruscolo. *Using NMR Chemical Shifts as Structural Restraints in Molecular Dynamics Simulations of Proteins.* Structure, vol. 18, no. 8, pages 923–933, 2010.

[Rodriguez 06] S. Rodriguez, X. Tang, J.M. Lien & N.M. Amato. *An obstacle-based rapidly-exploring random tree.* In Proceedings of the IEEE International Conference on Robotics and Automation, pages 895–900, 2006.

[Rohl 04] C.A. Rohl, C.E.M. Strauss, K. Misura & D. Baker. *Protein structure prediction using Rosetta.* Methods in enzymology, vol. 383, pages 66–93, 2004.

[Rossmann 76] M.G. Rossmann & P. Argos. *Exploring structural homology of proteins.* Journal of molecular biology, vol. 105, no. 1, pages 75–95, 1976.

[Sánchez 03] G. Sánchez & J.C. Latombe. *A single-query bi-directional probabilistic roadmap planner with lazy collision checking.* Robotics Research, pages 403–417, 2003.

[Schlick 10] T. Schlick. Molecular modeling and simulation: an interdisciplinary guide, volume 21. Springer Verlag, 2010.

[Schwartz 83] J.T. Schwartz & M. Sharir. *On the Òpiano movers'Ó problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers.* Communications on pure and applied mathematics, vol. 36, no. 3, pages 345–398, 1983.

[Schwede 08] T. Schwede & M.C. Peitsch. Computational structural biology: Methods and applications. World Scientific Publishing Company, 2008.

[Sciavicco 01] L. Sciavicco & B. Siciliano. Modelling and control of robot manipulators. Advanced Textbooks in Control and Signal Processing. Springer, 2001.

[Scott 66] R.A. Scott & H.A. Scheraga. *Conformational analysis of macromolecules. II. The rotational isomeric states of the normal hydrocarbons.* Journal of Chemical Physics, vol. 44, page 3054, 1966.

[Selkoe 03] D.J. Selkoe. *Folding proteins in fatal ways.* Nature, vol. 426, no. 6968, pages 900–904, 2003.

[Shaw 10] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan & W. Wriggers. *Atomic-level characterization of the structural dynamics of proteins.* Science, vol. 330, no. 6002, pages 341–346, 2010.

[Shehu 10] A. Shehu & B. Olson. *Guiding the Search for Native-like Protein Conformations with an Ab-initio Tree-based Exploration.* International Journal of Robotics Research, vol. 29, no. 8, pages 1106–1127, 2010.

[Siciliano 08] B. Siciliano & O. Khatib. Springer handbook of robotics. Gale virtual reference library. Springer, 2008.

[Simeon 00] T. Simeon, J.P. Laumond & C. Nissoux. *Visibility-based probabilistic roadmaps for motion planning.* Advanced Robotics, vol. 14, no. 6, pages 477–493, 2000.

[Singh 99] A.P. Singh, J.C. Latombe & D.L. Brutlag. *A Motion Planning Approach to Flexible Ligand Binding.* In Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, pages 252–261. AAAI Press, 1999.

[Song 02] G. Song & N.M. Amato. *Using motion planning to study protein folding pathways.* Journal of Computational Biology, vol. 9, no. 2, pages 149–168, 2002.

[Song 03] G. Song, S. Thomas, KA Dill, JM Scholtz & NM Amato. *A path planning-based study of protein folding with a case study of hairpin formation in protein G and L.* In Pacific Symposium on Biocomputing, pages 240–251, 2003.

[Soss 03] M. Soss, J. Erickson & M. Overmars. *Preprocessing chains for fast dihedral rotations is hard or even impossible.* Computational Geometry, vol. 26, no. 3, pages 235–246, 2003.

[Sousa 06] S.F. Sousa, P.A. Fernandes & M.J. Ramos. *Protein–ligand docking: current status and future challenges.* Proteins: Structure, Function, and Bioinformatics, vol. 65, no. 1, pages 15–26, 2006.

[Spong 06] M.W. Spong, S. Hutchinson & M. Vidyasagar. Robot modeling and control. John Wiley & Sons, 2006.

[Sternberg 96] M.J.E. Sternberg. Protein structure prediction: A practical approach. The Practical Approach Series. IRL Press at Oxford University Press, 1996.

[Şucan 09] I. Şucan & L. Kavraki. *Kinodynamic motion planning by interior-exterior cell exploration.* Algorithmic Foundation of Robotics VIII, vol. 57, pages 449–464, 2009.

[Sugita 99] Y. Sugita & Y. Okamoto. *Replica-exchange molecular dynamics method for protein folding.* Chemical Physics Letters, vol. 314, no. 1-2, pages 141–151, 1999.

[Tama 01] F. Tama & Y.H. Sanejouand. *Conformational change of proteins arising from normal mode calculations.* Protein Engineering, vol. 14, no. 1, pages 1–6, 2001.

[Tama 04] F. Tama, O. Miyashita & C.L. Brooks Iii. *Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-EM.* Journal of structural biology, vol. 147, no. 3, pages 315–326, 2004.

[Tang 05] X. Tang, B. Kirkpatrick, S. Thomas, G. Song & N.M. Amato. *Using motion planning to study RNA folding kinetics.* Journal of Computational Biology, vol. 12, no. 6, pages 862–881, 2005.

[Tang 08] X. Tang, S. Thomas, L. Tapia, D.P. Giedroc & N.M. Amato. *Simulating RNA folding kinetics on approximated energy landscapes.* Journal of Molecular Biology, vol. 381, no. 4, pages 1055–1067, 2008.

[Tapia 07] L. Tapia, X. Tang, S. Thomas & N.M. Amato. *Kinetics analysis methods for approximate folding landscapes.* Bioinformatics, vol. 23, no. 13, pages 539–548, 2007.

[Tapia 10] L. Tapia, S. Thomas & N.M. Amato. *A Motion Planning Approach to Studying Molecular Motions.* Communications in Information & Systems, vol. 10, no. 1, pages 53–68, 2010.

[Taylor 05] J.R. Taylor. Classical mechanics. University Science Books, 2005.

[Tenenbaum 00] J.B. Tenenbaum, V. Silva & J.C. Langford. *A global geometric framework for nonlinear dimensionality reduction.* Science, vol. 290, no. 5500, pages 2319–2323, 2000.

[Teodoro 01] M.L. Teodoro, G.N. Phillips Jr & L.E. Kavraki. *Molecular docking: A problem with thousands of degrees of freedom.* In Proceedings of the IEEE International Conference on Robotics and Automation, volume 1, pages 960–965, 2001.

[Thomas 05] S. Thomas, G. Song & N.M. Amato. *Protein folding by motion planning.* Physical biology, vol. 2, pages 148–155, 2005.

[Thomas 07] S. Thomas, X. Tang, L. Tapia & N.M. Amato. *Simulating protein motions with rigidity analysis.* Journal of Computational Biology, vol. 14, no. 6, pages 839–855, 2007.

[Thorpe 99] M.F. Thorpe & P.M Duxbury. Rigidity theory and applications: edited by m.f thorpe and p.m. duxbury. Springer US, 1999.

[Tirion 96] M. M. Tirion. *Large amplitude elastic motions in proteins from a single-parameter, atomic analysis.* Physical Review Letters, vol. 77, no. 9, pages 1905–1908, 1996.

[Tozzini 05] V. Tozzini. *Coarse-grained models for proteins.* Current Opinion in Structural Biology, vol. 15, no. 2, pages 144–150, 2005.

[Tsianos 07] Konstantinos I. Tsianos, Ioan Alexandru Sucan & L. E. Kavraki. *Sampling-based robot motion planning: Towards realistic applications.* Computer Science Review, vol. 1, pages 2–11, August 2007.

[Unger 93] R. Unger & J. Moult. *Genetic algorithms for protein folding simulations.* Journal of Molecular Biology, vol. 231, no. 1, pages 75–81, 1993.

[van den Bergen 98] G. van den Bergen. *Efficient collision detection of complex deformable models using AABB trees.* Journal of Graphics Tools, vol. 2, no. 4, pages 1–13, 1998.

[van der Maaten 09] L.J.P. van der Maaten, E.O. Postma & H.J. van den Herik. *Dimensionality Reduction: A Comparative Review.* Rapport technique TiCC-TR 2009-005, Tilburg University, 2009.

[Wallin 03] S. Wallin, J. Farwer & U. Bastolla. *Testing similarity measures with continuous and discrete protein models.* Proteins: Structure, Function, and Bioinformatics, vol. 50, no. 1, pages 144–157, 2003.

[Wells 05] S. Wells, S. Menor, B. Hespenheide & M. F. Thorpe. *Constrained geometric simulation of diffusive motion in proteins.* Physical Biology, vol. 2, pages 127–136, 2005.

[Wilmarth 02] S.A. Wilmarth, N.M. Amato & P.F. Stiller. *MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space.* In Proceedings of the IEEE International Conference on Robotics and Automation, volume 2, pages 1024–1031, 2002.

[Woolfson 97] M. M. Woolfson. An introduction to x-ray crystallography. Cambridge University Press, 1997.

[Wu 99] N. G. Wu & M. W. Deem. *Analytical Rebridging Monte Carlo: Application to cis/trans Isomerization in Proline-Containing, Cyclic Peptides.* Journal of Chemical Physics, vol. 111, pages 6625–6632, 1999.

[Xie 03] M. Xie. Fundamentals of robotics: Linking perception to action. Series in Machine Perception and Artificial Intelligence. World Scientific Pub., 2003.

[Yang 07] H. Yang, H. Wu, D. Li, L. Han & S. Huo. *Temperature-dependent probabilistic roadmap algorithm for calculating variationally optimized conformational transition pathways.* J. Chem. Theory Comput, vol. 3, no. 1, pages 17–25, 2007.

[Yao 08] P. Yao, A. Dhanik, N. Marz, R. Propper, C. Kou, G. Liu, H. van den Bedem, J. Latombe, I. Halperin-Landsberg & R. B. Altman. *Efficient Algorithms to Explore Conformation Spaces of Flexible Protein Loops.* IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 5, pages 534–545, 2008.

[Zaki 08] M.J. Zaki & C. Bystroff. Protein structure prediction. Methods in Molecular Biology. Humana Press, 2008.

[Zhang 02] M. Zhang & L.E. Kavraki. *A new method for fast and accurate derivation of molecular conformations.* Journal of Chemical Information and Computer Sciences, vol. 42, no. 1, pages 64–70, 2002.

[Zhao 94] Jianmin Zhao & Norman I. Badler. *Inverse kinematics positioning using nonlinear programming for highly articulated figures.* ACM Trans. Graph., vol. 13, no. 4, pages 313–336, 1994.