Motion Planning Algorithms for Molecular Simulations: A Survey

Ibrahim Al-Bluwi[a,b], Thierry Siméon[a,b], Juan Cortés[a,b]

[a] *CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse Cedex 4, France*
[b] *Université de Toulouse; UPS, INSA, INP, ISAE; UT1, UTM, LAAS; F-31077 Toulouse Cedex 4, France*

Abstract

Motion planning is a fundamental problem in robotics that has motivated active research from more than three decades ago. A large variety of algorithms has 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 papers concerns the latter domain, providing (up to our knowledge) the first overall survey on motion planning algorithms applied to molecular modeling and simulation. This paper discusses both the algorithmic and application sides of these methods 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 paper gives a general overview on the different extensions of sampling-based motion planners that proposed in this context. From the point of view of applications, the survey deals with problems involving protein folding and conformational transitions, as well as protein-ligand interactions.

Keywords: Motion Planning, Sampling-based Algorithms, Molecular Simulations, Protein Flexibility, Protein Folding, Protein-Ligand Interactions.

1. Introduction

Nowadays, computer simulations are widely used to model biomolecules, mimic their behavior and gain insight about their physiochemical properties

and biological functions. Indeed, a whole field dedicated to such simulations currently exists under the name of Computational Structural Biology.

The need for such molecular simulations mainly stems from the limitations of current experimental methods. For example, determining three-dimensional structures of proteins can be achieved experimentally using methods like X-Ray Crystallography [1] and Nuclear Magnetic Resonance (NMR) [2]; however, such methods suffer from several shortcomings. These methods provide only static or averaged information about the structure under study, which is insufficient as proteins are known to be flexible and dynamic. Computational methods have been developed for complementing, and even for replacing experimental methods. For instance, methods such as Molecular Dynamics (MD) [3] and Monte Carlo (MC) methods [4] are largely used to study thermodynamic properties and activity of proteins from an initial structure determined by X-Ray crystallography or NMR. Other computational methods can be used to determine the structure of proteins without prior experimental information [5]. They are also used for predicting molecular interactions (Molecular Docking) [6], and for understanding how proteins move from random coils to their native structure (Protein Folding) [7]. Nevertheless, the current status of these computational methods is still far for 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 Molecular Dynamics 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 a few seconds [8]. On the other hand, Monte Carlo (MC) simulations also suffer from shortcomings in their search and sampling of the conformational space of proteins, which is a rugged landscape that is full of 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 [9, 10, 11] for examples) and producing alternatives for them. This paper surveys a particular family of such alternative methods that are inspired from the field of robot motion planning. Robotics-inspired methods have been introduced recently for simulating motions of proteins and studying problems like protein folding and protein-ligand interactions. They borrow ideas from sampling-based motion planning algorithms [12, 13], which have proven to be very successful in tackling high-dimensional robot

motion planning problems.

Although the two fields of robotics and molecular simulations seem very distant at first glance, a look under the hood reveals many similarities in terms of the formulation of the tackled problems. In an early survey [14], 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. Robots cannot be called robots unless they move; they are otherwise simply computers or electronic devices. Similarly, protein motions are 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 understanding disfunctions and their contribution to diseases such as the mad cow disease and Alzheimer's disease [15].

Since motion-planning-inspired algorithms for molecular simulations are relatively new, up to our knowledge, no dedicated reviews have been written on this subject. Nevertheless, there are two works that are noteworthy in this regard. The first is a survey by Moll *et al.* [16] that is dedicated to applications of motion planning roadmap methods to protein folding. The second is an online course prepared by Kavraki entitled "Geometric Methods in Structural Computational Biology"[17]. 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 implementations rather than surveying the current literature; which is the aim of this paper.

The paper is structured as follows: Section 2 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. Main molecular simulation methods that are inspired by robot motion planning are then surveyed and explained in Section 3. Next, Section 4 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 5 summarizes and concludes the survey.

2. 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) [18] and Rapidly-Exploring Random Trees (RRT) [19], 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.

2.1. Motion Planning in Robotics

Robot Motion Planning consists in deciding automatically what motions a robot should execute in order to achieve a task specified by initial and goal spatial arrangements of physical objects [20]. A frequently used example to explain the idea is as follows: 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 [21].

Motion planning is generally formulated using the notion of Configuration Space [22]. 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 now a problem of 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 [23]. Some examples of such methods are: Cell Decomposition [24], Visibility Graphs [25] and Voronoi-Diagram-based methods [26]. These methods construct roadmaps that are guaranteed to cover the whole space. The problem, however, is that they rely on an explicit representation of C_{obs}, which can be difficult to construct, especially for high-dimensional configuration spaces. Indeed, it has been shown that finding a complete solution for the

motion planning problem in general is PSPACE-hard [27], and that, even if a roadmap is constructed, finding the shortest path between any two vertices in a three dimensional configuration space is NP-complete [28]. For this reason, attention has shifted to practical motion planning algorithms rather than complete ones. Sampling-Based Motion Planners [29, 30, 12] are one type of such 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 [12].

Sampling-based planners sample the configuration space to build a representative set of configurations, which substitutes for the explicit representation of the configuration space. The difference between a planner and another lies mainly in how sampling is performed and how the samples are connected. Sampling-based planners are often classified into two categories: *Roadmap-Based Planners* and *Tree-Based Planners*. Roadmap methods basically work in two phases: a construction phase and a query phase. In the construction phase, a graph that covers the configuration space is built and in the query phase this graph is used to plan the motion between any needed start and goal configurations. 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-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 and RRT, are the most representative methods of each of these main classes.

2.1.1. Probabilistic Roadmap

The Probabilistic Roadmap (PRM) algorithm was introduced in the nineties [18] and was a breakthrough. It was able to successfully solve motion planning problems with higher dimensions than what was achieved before. The basic version of PRM[1] 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

[1]This is one of the basic variants of the algorithm. Another version performs sampling and connections in separate loops.

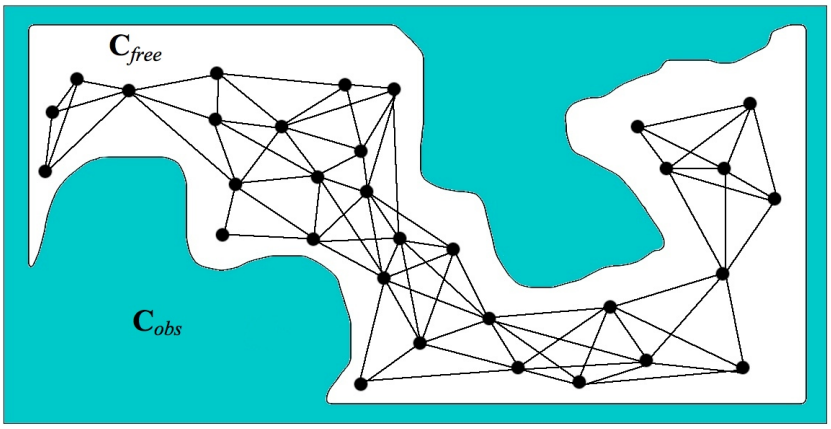

Figure 1: An illustration of a simple PRM.

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 straight line[2]. 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. The graph at hand can then be searched for paths using any of the conventional graph search algorithms such as Dijkstra's shortest path and the A* algorithms. These basic steps of the PRM have been improved over the years and several variants have appeared (e.g. [31, 32, 33, 34]). However, the general structure of the algorithm remains the same. Figure 1 shows an illustrative example of the basic PRM.

2.1.2. Rapidly Exploring Random Tree

The most popular tree-based motion planner is the Rapidly-Exploring Random Tree (RRT) [19]. Rooted at the start configuration, this tree grows 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

[2]This applies for systems without differential constraints and which admit any interpolation between a pair of states. Under such constraints, more complex methods are required to perform local connections.

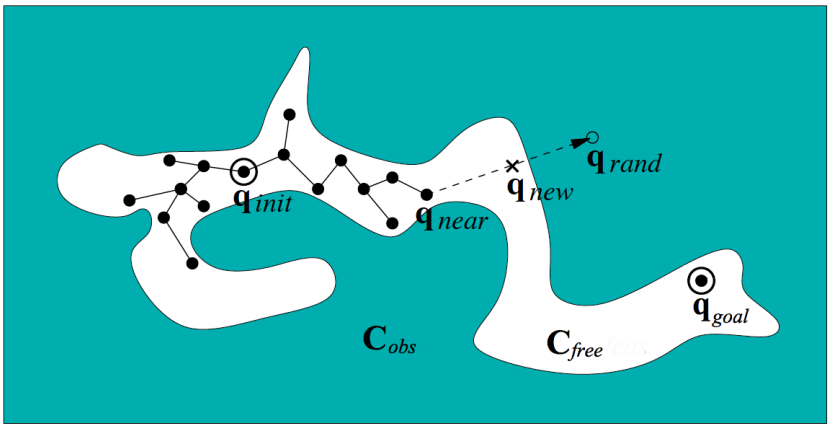

Figure 2: Illustration of a simple RRT at an intermediate stage during its construction. Here, q_{init} and q_{goal} are the initial and goal configurations, respectively.

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 extension, 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 (this applies for systems without differential constraints and which admit any interpolation between a pair of states):

1. A random configuration q_{rand} is sampled from 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 interpolating on the straight line between q_{near} and q_{rand} with a predefined distance d.
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 repeats until the goal configuration can be connected to the tree or a maximum number of iterations is reached. Figure 2 shows an illustrative example of the basic RRT algorithm. Variants of this basic algorithm appeared later on (e.g. [35, 36, 37, 38]). Moreover, other tree-based planners that are not directly based on RRT have also been proposed. Two examples of such planners are the Expansive Spaces Tree [39] and the Path-Directed Subdivision Tree [40].

2.2. 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 of studying 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.

2.2.1. Molecular Representation

The most straightforward way for representing molecules geometrically is to list the Cartesian coordinates of all the atoms. Bonds can then be constructed automatically using the distances between atoms and the knowledge about their types. This representation is called the *Cartesian representation* and it is used by the Protein Data Bank [41] 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 3 for an illustration. Although bond lengths and bond angles vary, their variation is known to be very small at room temperature, to an extent that it is possible to ignore them. On the other hand, major conformational changes in the molecule occur due to variations in dihedral angles. For this reason, it is commonly assumed that dihedral

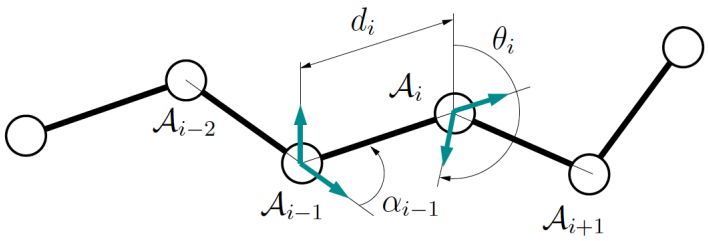

Figure 3: Parameters defining the relative position of bonded atoms.

angles are the only degrees of freedom of the molecule[3]This is usually called the rigid geometry assumption [42].). Hence, the conformation of a molecule can be represented as a vector of dihedral angles. This representation is called the *internal coordinates representation* and it has been adopted by most motion-planning-inspired algorithms as it corresponds to how articulated robots are represented. Figure 4 shows a protein model together with a representation of dihedral angles corresponding to one of its amino acid residues. Note 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 [43].

2.2.2. 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 4.3), 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 [44]. This problem of high dimensionality is actually one of the fundamental 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

[3](

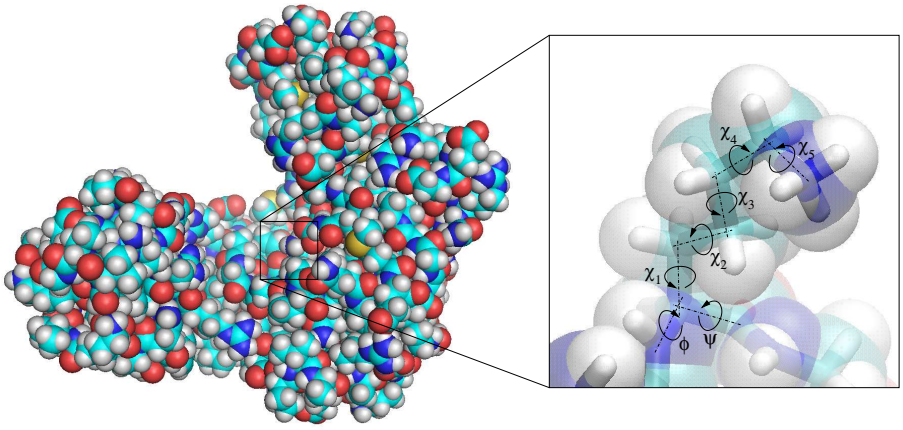

Figure 4: The main image shows a protein model in van der Waals representation (spheric atoms). The detail shows one its constituent amino acid residues and the dihedral angles required to define its conformation.

that the receptor protein is rigid [45]. However, it has been shown that this assumption renders the achieved solutions unrealistic since receptors may go through important conformational changes [46]. Other works have made more realistic assumptions based on prior chemical knowledge of the receptor protein. Using this knowledge, dihedral angles that can 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 with such methods is that they are problem-dependent and hard to automate [44]. A more general approach proposed in [47] consists in choosing automatically which parts of the protein can be considered as rigid bodies and which parts have to be considered as flexible using methods based on rigidity theory [48, 49]. Another strategy to reduce the problem dimensionality is to assume that secondary structure elements are rigid, and that loops, linkers and side-chains are flexible. This approach, as in [50], 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 [51, 52] to map the current degrees of freedom into a lower-dimensional space. These methods usually begin with a priorly 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 Principle Component Analysis (PCA) [53], which is commonly used in the analysis of near-equilibrium fluctuations sampled by molecular dynamics simulations [54]. However, PCA may not be suitable for 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 [55]. 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 [54]. This method was further extended in [56] to be even more efficient by performing distance measures in yet another projection on a lower dimensional Euclidean space.

Normal Mode Analysis [57] has also been used in this regard. It has been shown that large-amplitude motions in proteins are related to low-frequency normal modes [58, 59]. Consequently, low-frequency normal modes can be used to predict the direction of large-amplitude motions. In [60], transition pathways between conformations are computed using an RRT-like algorithm that explores linear combinations of low-frequency normal modes. An advantage of such a method over methods like PCA and IsoMap is that normal modes are computed online and no data set of conformations is required to be available a priori.

2.2.3. Distance Metrics

In molecular simulations, we often need 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 2.1, RRT-based methods rely on finding the most similar conformation to every new random sample. PRMs also searches for local connections between neighbor nodes corresponding to similar conformations. This makes the choice of the distance measure very influential on the performance of the whole algorithm.

A widely used and straightforward distance measure is the coordinate root mean squared deviation (cRMSD). If two molecular conformations are represented as vectors of the Cartesian coordinates of their atoms, then cRMSD

is the square root of the average squared distance between the corresponding atoms. This requires both conformations be aligned in order to remove the effect of any translation or rotation of the whole molecule. 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. It is also possible to apply the root mean squared deviation 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 here that in molecular simulations we are more interested in distance measures that capture structural differences in proportion with their effect on the potential energy of the molecule. 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.

Since computing distances can be a bottleneck for motion-planning-inspired methods, especially if all-atom measures like dRMSD and cRMSD are used, 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 appear in the literature. One example is the work of Lotan and Schwarzer [61], 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 [62]. 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 [63] for the problem of molecular disassembly (see section 4.3), 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, which could be applied for molecular simulations, is proposed in [64]. This method projects the sampled conformations S to an m dimensional Euclidean space and performs the distance measures in that space. The projection is done by first selecting m pivots from S and then replacing each variable s in S by a vector of the distances between s 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.

2.2.4. 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 2.1, 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 collisions 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 [65, 66] and several general-purpose collision detection packages are available (e.g. [67, 68, 69]). 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 [70, 71] 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 [72] 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 [73] was specifically designed to be used within sampling-based motion planning algorithms applied on proteins described as kinematic chains. It assumes that only a pre-selected set of the protein degrees of freedom 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 interaction does not change from one iteration to another.

2.2.5. Treating 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 interactions [74]. 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. [75, 76]). Note however that most of such methods are limited to 6 degreed of freedom, and therefore, extensions are necessary to deal with long loops. In [77], 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 [78], 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 [79], 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. A notable example of such methods is the Cyclic Coordinate Descent (CCD) [80]. Given the start and end points, CCD samples an open conformation for the chain segment rooted at the start point, and then iteratively adjusts one dihedral

angle at a time in a way that minimizes the distance between the end frame in the sampled conformation and the required end point.

2.2.6. Energy Computation

As mentioned in Section 2.2.1, 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 [81]; 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 [82] is usually used to provide approximate energy values of protein conformations. These values do not make much sense when read alone, but are very useful when read in comparison to each other. 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 [83, 84] 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 a *Coarse Grained* force fields [85]. 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 [86] and OPEP [87].

3. Motion-Planning-Inspired Methods for Molecular Simulations

Seminal work on the application of motion planning algorithms to the study of proteins was published in 1999 [88]. 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.

3.1. PRM-Based Methods

3.1.1. Probabilistic Conformational Roadmaps

The method proposed by Singh *et al.* [88] 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 be different from the conventional PRM in robotics that uses collision detection for evaluating new samples. The used probability function is as follows:

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

where E_q is the potential energy of conformation q, and E_{min} and E_{max} are threshold values that depending on the molecular system in hand. Neighboring nodes are then connected, and a weight is associated to each edges. These weights are actually 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}:

$$P_i = \frac{e^{-(E_{i+1}-E_i)/KT}}{e^{-(E_{i+1}-E_i)/KT} + e^{-(E_{i-1}-E_i)/KT}}, \tag{2}$$

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, where extra nodes are sampled 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 [89], this method was given the name of Probabilistic Conformational Roadmaps (PCR), and was applied to study protein folding.

3.1.2. Stochastic Roadmap Simulations

Stochastic Roadmap Simulations (SRS) [90, 91, 92, 93, 94] 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 [95], which allows for establishing a connection between SRS and Monte Carlo methods. The transition probability used in SRS is as follows:

$$P_{ij} = \begin{cases} \frac{1}{n_i} \exp(-\frac{\triangle E_{ij}}{KT}) & if \triangle E_{ij} > 0 \\ \frac{1}{n_i} & otherwise \end{cases} \tag{3}$$

$$P_{ii} = 1 - \sum_{j \neq i} P_{ij}, \tag{4}$$

where $\triangle E_{ij}$ is the difference in potential energy between nodes q_i and q_j, n_i is the number of neighbors to q_i. $\varepsilon = \exp(-E/KT)$ is the Boltzmann factor. 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 4.2). Every path in the roadmap can be looked at as the run of Markov Chain Monte Carlo (MCMC) method. This allows for interpreting the whole roadmap as the result of a set of MCMC being run simultaneously. In fact in [91], 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 with the cost of possibly loosing important information along the paths between nodes.

3.1.3. PRMs for Folding Pathways

Another research line that started early is the work led by Nancy Amato [96, 97, 98, 99, 100, 101, 47, 102, 103]. 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). 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} [103]. This method also uses the following formula for edge weights, which is a slightly modified version of equation 2:

$$P_i = \begin{cases} e^{\frac{-\triangle E_i}{KT}} & if \ \triangle E_i > 0 \\ 1 & if \ \triangle E_i \leq 0 \end{cases} \tag{5}$$

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 concerned 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 [96, 97], 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 [98, 99] another strategy was proposed since the first one worked well only for proteins with around 60 residues or less. 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 [47], 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 by other researchers derived from this method have been proposed more recently. An example is the MaxFlux-PRM [104, 105], which uses a different edge weight function in order to find optimal reaction paths that are temperature-dependent.

3.2. RRT-Based Methods

3.2.1. Manhattan-Like RRT

Manhattan-Like RRT (ML-RRT) is a variant of the RRT algorithm proposed in [106] for treating a special case of the motion planning problem called *(dis)assembly path planning*. The problem consists in 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 5 for an illustration).

ML-RRT works by dividing configuration parameters into two groups, called active and passive, and by generating 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 an 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, but not less important, 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 was able to solve efficiently problems involving the flexibility of the ligand and the protein side chains. The extensions proposed in [50] enables 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.

3.2.2. Transition-RRT

Another RRT variant called Transition-RRT (T-RRT) was introduced in [107, 108] 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

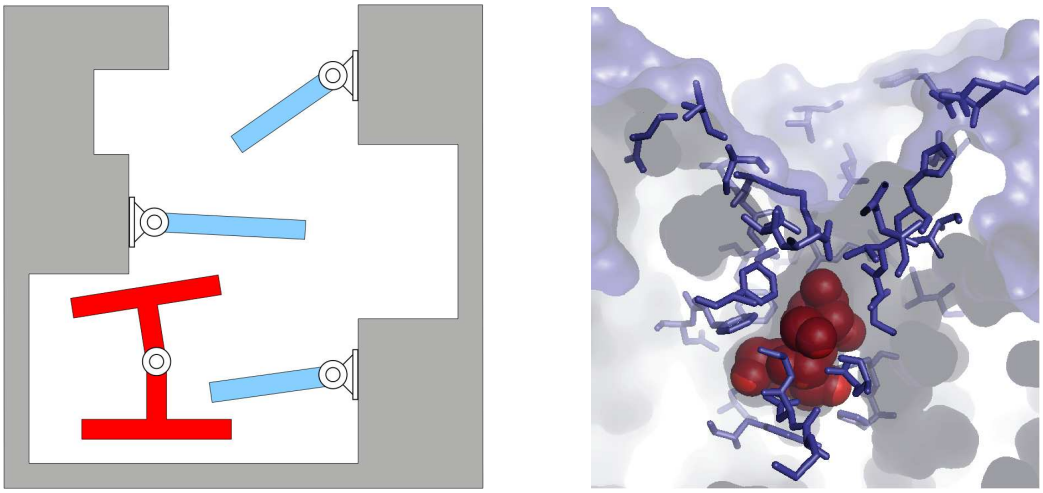

Figure 5: The image on the left illustrates an academic disassembly planning problem for two articulated objects. An analogy can be made with the protein-ligand "disassembly" problem represented in the right-hand image. The red object can be considered as the ligand and the blue sticks as flexible side-chain of the protein.

and added to the tree with the following probability :

$$P_{ij} = \begin{cases} e^{\frac{-\triangle E_{ij}}{KT}} & if\ \triangle E_{ij} > 0 \\ 1 & otherwise \end{cases} \tag{6}$$

In this equation, $\triangle E_{ij}$ is the difference between the energy at q_{near} and at q_{new}. 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 node insertion 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 is decreases, therefore increasing the severity of the transition test. Thus, the temperature is automatically regulated along 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.

3.2.3. NMA-RRT

The work by Kirillova *et al.* [60] proposes an RRT-based methods that applies Normal Mode Analysis (NMA) [57] for computing global macromolecular motions. As mentioned in Section 2.2.2, 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 fact is put into use 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 constrains 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.

3.2.4. PathRover

A simulation framework named *PathRover* was presented in [109] for sampling and generating motion pathways between molecular conformations. It uses the RRT algorithm and applies a *branch-termination* scheme to satisfy 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). This RRT variant also uses a validity test based on the energy of the conformations. All conformations with an energy value higher than a given threshold are considered to fall in C_{obs}, whereas all the rest are considered to fall in C_{free}.

Note that PathRover is an extension of earlier work by the same group [110] on the computation of conformational transition pathways of proteins.

3.3. Other Methods

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

In [62], Shehu *et al.* proposed a tree-based method called *FeLTr* for studying the problem of protein structure prediction (see Section 4.2). This method uses a tree structure 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 more refined search to obtain the folded conformation. FeLTr uses a coarse grained representation of the protein and a two-layered search strategy that tries to sample low energy conformations without oversampling geometrically similar ones.

Another motion-planning-based method was introduced in [111] for computing large-amplitude motions between molecular conformations. This method is based on the Path Directed Subdivision Tree (PDST) algorithm [40], 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 [40]. In order to enhance the performance of the method, a coarse-grained protein model and a simplified energy function were considered. The distance metric was defined in terms of the relative positions between the secondary structure elements.

4. Applications

The methods presented in the previous section have been mainly applied to three types of problems in computational structural biology: the computation of conformational transitions of proteins, the study of the protein folding precess, and the analysis of protein-ligand interactions. This section discusses briefly each of these problems and present the main results achieved by motion-planning-inspired methods.

NOTE: ADD FIGURES ILLUSTRATING THE PROBLEMS

4.1. Conformational Transitions

The most direct application for robot motion planning methods in molecular simulations consists in computing transition pathways between two molecular conformations. This problem requires generating a sequence of feasible intermediate conformations for the molecule (usually a protein) to link the

given conformations. The problem is analogous to the motion planning problem in robotics. Computing conformational transition pathways if proteins is important for understanding their biological functions. This problem can be seen as a general instance of several more particular 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 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 phenomena under study. Some phenomena 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). For such phenomena, the problem can be tackled at a structural level that is higher than the level of individual atoms. In other cases interest may be focused on flexible segments of the protein. For example, irregular segments in the protein, 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 6 illustrates these two types of protein motions.

4.1.1. Loop Motions

Results on the application of an RRT-based algorithm extended to treat closed kinematic chains (RLG-RRT) [77] for computing protein loop motions were first presented in [113]. The algorithm was tested on *Amylosucrase* (*AS*), considering loop 7 as an articulated mechanism and the rest of the protein as a rigid body. Results were positive, as they showed the effectiveness of motion-planning-based methods for studying the mobility of loops. An improved version of the method, which integrates ideas of ML-RRT, was applied in [114] to investigate the large-scale open-to-closed movement of the lid that controls the access to the active site of *Burkholderia cepacia* lipase (*BCL*). Results show that the lid conformational transition computed with

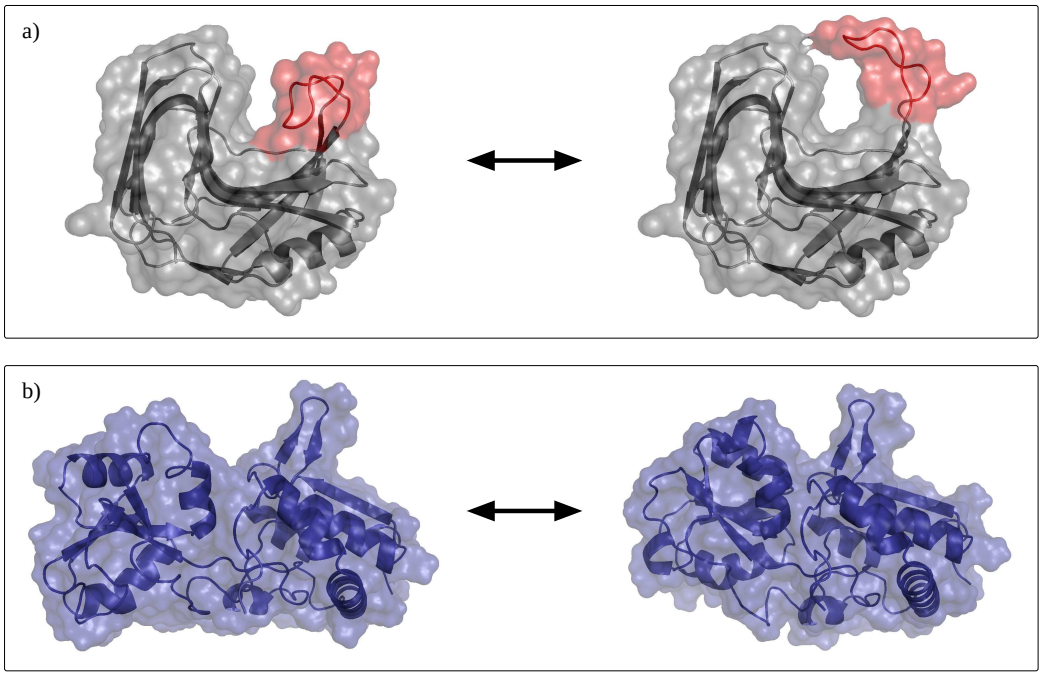

Figure 6: Illustration of two classes of large-amplitude motions in proteins. (a) Loop motions: A segment of the protein (in red) moves significantly, while the rest of the protein remains mostly static. (b) Domain motions: Large portions of the protein move with respect to each other.

this method is comparable to the one obtained with molecular dynamics simulations. Nevertheless, the computing time requited by the RRT-based method is several orders of magnitude lower.

Several tests on the application of LoopTK to study motions for 20 different loops are presented in [79]. Results show that LoopTK can sample efficiently 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.

4.1.2. Domain Motions

The results reposted in [60] 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 benchmark. The abbreviated names of these proteins are: *ADK*, *ATP*, *DAP*, *EIA* and *LAO*. Tests showed that NMA-RRT produces results that correlate well with previous studies. NMA-RRT was able to achieve these results using a notably 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 [109]. The particular phenomenon studied in these tests is *domain swapping*, and the achieved results are consistent with experimental results. Moreover, in [110], 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.

Conformational transition simulations have also been performed using the PDST-based method presented in [111]. 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 [112]. The paper also shows that results of the PDST-based method are consistent with experimental data.

4.2. Protein Folding

Protein folding is the process in which proteins move (fold) from random coils to their native three-dimensional shape. Being in the correct folded state is essential for proteins to function properly in many biological processes, and, usually, unfolded or incorrectly folded proteins are inactive or even toxic [115, 15]. 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 [116], 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 [117, 118], Ultrarapid Mixing [119] and Time-Resolved Absorption Spectroscopy [120]. However, the information obtained from these methods is currently limited. Computational methods have been used side by side with these experimental methods, either augmenting them or even replacing them (for examples, see [121, 9, 122, 123]). Important advances with these computational methods started with the advent of the energy landscape theory [124], which hypothesizes that the energy landscape

of a protein is funneled with many pathways 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 advantages of such methods over most of conventional methods are their ability to rapidly explore the conformational space without getting trapped in local energy minima, and their capacity to find several pathways simultaneously.

4.2.1. Computation of Folding Quantifiers

The Stochastic Roadmap Simulations (SRS) method has been used in the computation of different types of quantifiers and ensemble properties related to protein folding: the probability of folding (P_{fold}), the Transition State Ensemble (TSE), the folding rate, and the Φ-value of residues. P_{fold} is the probability that the structure at a certain conformation would become completely folded before it becomes completely unfolded. TSE 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 measures the ratio of proteins in a set that advance towards the folding state per unit time. The Φ-value measures how close a certain residue is to its native folded state.

In [91, 92], 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 [93, 94], where 16 proteins were analyzed using SRS to compute TSEs, folding rates and Φ-values. Results were then compared to 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 [101] and used in combination with the conformational explo-

ration method presented in Section 3.1.3 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 mention that the capacity of FeLTr to predict native-like conformations of small-to-medium size proteins has been shown in [62]. 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.

4.2.2. 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 [96, 97, 98, 99]. 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. The 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 [97] 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 [47].

4.2.3. RNA (Un)folding Pathways

The combination of the PRM-based exploration with MME and MMC discussed above has also been used in [100, 102] to study the problem of RNA (un)folding, which is a problem 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.

4.3. Protein-Ligand Interactions

The study of protein-ligand interactions is indispensable 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 elements to be studied

are the way the protein recognizes a particular ligand, how the ligand binds the protein active site, and what conformational changes both molecules undergo during the ligand's entrance to the active site or its exit from it. Such information allows predicting the possibility of association between protein-ligand pairs, the strength of this association, or the protein activity level. However, obtaining atomic-scale information about protein-ligand interactions using current experimental methods is practically infeasible. Moreover, the largeness 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 [125], DOCK [126], FleX [127], GOLD [128] and ICM [129]. These packages use algorithms such as Monte Carlo, Molecular Dynamics, Genetic Algorithms [130], and fragment-based search [131] (for a survey of methods and software packages see [132]). However, none of these softwares 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. Next, we survey works that use motion-planning-inspired methods for predicting binding sites and for computing access/exit ligand pathways.

4.3.1. Predicting Binding Sites

The algorithm of Singh *et. al.* introduced in [88] was tested on the following three protein-ligand complexes: *Lactate Dehydrogenase* with *Oxamate*, *tyrosyl-transfer-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 a 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 [90], 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 [90]. Tests were performed on seven different protein-ligand complexes and results showed that in five out of seven complexes, 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 [88], only the ligand was assumed to be flexible and the protein was assumed to be rigid. This is possibly one of the reasons to explain why these methods fail in some cases.

4.3.2. Finding Access and Exit Pathways

The RRT-based method presented in [78] 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 pseudo-molecular dynamics. However, the combined RRT-minimization approach only required some 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 a highly valuable information for site directed mutagenesis. Further investigations about the influence of ligand access/exit on *Burkholderia cepacia* lipase enantioselectivity are presented in [133, 134]. These works show the ability of RRT-based methods to produce results rapidly, which presented a fair qualitative agreement with experimental studies.

The extended ML-RRT method described in [50] was applied to compute the exit pathways of a bound substrate homologue (TDG) from *Lactose permease* (LacY) and of *carazolol* from the active site of the β_2-adrenergic receptor. 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.

5. 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. 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

Application Domain	Related Work
Loop Motions	**RLG-RRT** [113, 78, 114], **LoopTK** [79].
Domain Motions	**NMA-RRT** [60], **PathRover** [110, 109], **PDST** [111].
Protein Folding/Unfolding	**SRS** [90, 91, 92, 93, 94], **PRM** [96, 97, 98, 99, 101, 47, 103], **MaxFlux-PRM** [104, 105]
RNA Folding	**PRM** [100, 102].
Protein Structure Prediction	**FeLTr** [62].
Protein-Ligand Interactions	**PCR** [88, 89], **SRS** [90], **ML-RRT** [133, 134].

Table 1: Motion planning inspired methods classified according to application domain.

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 taken into account, and a choice has to be taken for the type of force field to be used.

Motion-planning-inspired methods for molecular simulations are still in their early stage. First results show that such methods are promising complementary methods to more conventional techniques in computational structural biology. Their strength lies mainly in their efficiency for exploring highly complex spaces. Yet, they still require 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. Researchers also need to look into other classes of problems than the ones already tackled in order to broaden the applicability of these methods. For instance, a particularly interesting problem that remains to be addressed with motion planning methods is protein-protein docking.

Our goal with this survey is twofold: (1) For readers in the structural biology community, we expect this paper will serve as an introduction to robotics-inspired methods with applications in their domain, and that this work will contribute to spreading this new family of methods in this com-

munity; (2) For readers in the robotics community, our aim is to incite them to look at problems in structural biology, which represent challenging benchmarks that motivate the improvement of algorithms.

References

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

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

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

[4] D. Landau, K. Binder, A guide to Monte Carlo simulations in statistical physics, Cambridge University Press, 2005.

[5] R. Bonneau, D. Baker, Ab initio protein structure prediction: progress and prospects, Annual Review of Biophysics and Biomolecular Structure 30 (1) (2001) 173–189.

[6] T. Lengauer, M. Rarey, Computational methods for biomolecular docking, Current Opinion in Structural Biology 6 (3) (1996) 402–406.

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

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

[9] Y. Sugita, Y. Okamoto, Replica-exchange molecular dynamics method for protein folding, Chemical Physics Letters 314 (1-2) (1999) 141–151.

[10] E. Marinari, G. Parisi, Simulated tempering: a new monte carlo scheme, Europhysics letters 19 (6) (1992) 451–458.

[11] A. Laio, M. Parrinello, Escaping free-energy minima, Proceedings of the National Academy of Sciences of the United States of America 99 (20) (2002) 12562–12566.

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

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

[14] D. Parsons, J. Canny, Geometric problems in molecular biology and robotics, in: Proceedings of the International Conference on Intelligent Systems for Molecular Biology, 1994, pp. 322–220.

[15] D. Selkoe, Folding proteins in fatal ways, Nature 426 (6968) (2003) 900–904.

[16] M. Moll, D. Schwarz, L. E. Kavraki, Roadmap Methods for Protein Folding, Humana Press, 2007.

[17] L. Kavraki, Geometric methods in structural computational biology. URL `http://cnx.org/content/col10344/1.6/`

[18] L. Kavraki, P. Svestka, J. Latombe, M. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE transactions on Robotics and Automation 12 (4) (1996) 566–580.

[19] 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, 2001, pp. 293–308.

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

[21] J. Schwartz, M. Sharir, On the piano movers' problem i. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers, Communications on pure and applied mathematics 36 (3) (1983) 345–398.

[22] T. Lozano-Peréz, Spatial planning: A configuration space approach, IEEE Transactions on Computers 32 (2) (1983) 108–120.

[23] K. Goldberg, Completeness in robot motion planning, in: Proceedings of the workshop on Algorithmic foundations of robotics, A. K. Peters, Ltd., Natick, MA, USA, 1995, pp. 419–429.

[24] B. Chazelle, Approximation and decomposition of shapes, Algorithmic and Geometric Aspects of Robotics (1987) 145–185.

[25] T. Lozano-Pérez, M. A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, Commun. ACM 22 (1979) 560–570.

[26] C. ó'Dúnlaing, M. Sharir, C. K. Yap, Retraction: A new approach to motion-planning, in: Proceedings of the fifteenth annual ACM symposium on Theory of computing, STOC '83, ACM, 1983, pp. 207–220.

[27] J. H. Reif, Complexity of the mover's problem and generalizations, in: Proceedings of the 20th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1979, pp. 421–427.

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

[29] S. Lindemann, S. LaValle, Current issues in sampling-based motion planning, Robotics Research (2005) 36–54.

[30] K. I. Tsianos, I. A. Sucan, L. E. Kavraki, Sampling-based robot motion planning: Towards realistic applications, Computer Science Review 1 (2007) 2–11.

[31] N. Amato, O. Bayazit, L. 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, 1998, pp. 155–168.

[32] T. Simeon, J. Laumond, C. Nissoux, Visibility-based probabilistic roadmaps for motion planning, Advanced Robotics 14 (6) (2000) 477–493.

[33] S. Wilmarth, N. Amato, P. 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, Vol. 2, 2002, pp. 1024–1031.

[34] G. Sánchez, J. Latombe, A single-query bi-directional probabilistic roadmap planner with lazy collision checking, Robotics Research (2003) 403–417.

[35] J. Kuffner Jr, S. LaValle, RRT-connect: An efficient approach to single-query path planning, in: Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 2, 2000, pp. 995–1001.

[36] J. Bruce, M. Veloso, Real-time randomized path planning for robot navigation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, 2002, pp. 2383–2388.

[37] P. Cheng, S. LaValle, Resolution complete rapidly-exploring random trees, in: Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, 2002, pp. 267–272.

[38] S. Rodriguez, X. Tang, J. Lien, N. Amato, An obstacle-based rapidly-exploring random tree, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2006, pp. 895–900.

[39] D. Hsu, J. Latombe, R. Motwani, Path planning in expansive configuration spaces, in: Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 3, 1997, pp. 2719–2726.

[40] A. M. Ladd, L. E. Kavraki, Fast Tree-Based Exploration of State Space for Robots with Dynamics, Springer, 2005, pp. 297–312.

[41] H. Berman, T. Battistuz, T. Bhat, W. Bluhm, P. Bourne, K. Burkhardt, Z. Feng, G. Gilliland, L. Iype, S. Jain, et al., The protein data bank, Acta Crystallographica Section D: Biological Crystallography 58 (6) (2002) 899–907.

[42] R. Scott, H. Scheraga, Conformational analysis of macromolecules. ii. the rotational isomeric states of the normal hydrocarbons, Journal of Chemical Physics 44 (1966) 3054.

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

[44] M. Teodoro, G. Phillips Jr, L. Kavraki, Molecular docking: A problem with thousands of degrees of freedom, in: Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, 2001, pp. 960–965.

[45] A. R. Leach, Molecular Modelling: Principles and Applications, Pearson Education, 2001.

[46] C. Cavasotto, A. Orry, R. Abagyan, The challenge of considering receptor flexibility in ligand docking and virtual screening, Current Computer-Aided Drug Design 1 (4) (2005) 423–440.

[47] S. Thomas, X. Tang, L. Tapia, N. Amato, Simulating protein motions with rigidity analysis, Journal of Computational Biology 14 (6) (2007) 839–855.

[48] M. Thorpe, P. Duxbury, Rigidity theory and applications: edited by M.F Thorpe and P.M. Duxbury, Springer US, 1999.

[49] S. Wells, S. Menor, B. Hespenheide, M. F. Thorpe, Constrained geometric simulation of diffusive motion in proteins, Physical Biology 2 (2005) 127–136.

[50] J. Cortés, D. Le, R. Iehl, T. Siméon, Simulating ligand-induced conformational changes in proteins using a mechanical disassembly method, Physical Chemistry Chemical Physics 12 (29) (2010) 8268–8276.

[51] I. K. Fodor, A survey of dimension reduction techniques, Tech. Rep. UCRL-ID-148494, Lawrence Livermore National Lab (June 2002).

[52] L. van der Maaten, E. Postma, H. van den Herik, Dimensionality reduction: A comparative review, Tech. Rep. TiCC-TR 2009-005, Tilburg University (2009).

[53] I. Jolliffe, Principal component analysis, Springer Verlag, 2002.

[54] P. Das, M. Moll, H. Stamati, L. Kavraki, C. Clementi, Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction, Proceedings of the National Academy of Sciences 103 (26) (2006) 9885–9890.

[55] J. Tenenbaum, V. Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[56] 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 67 (4) (2007) 897–907.

[57] 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.

[58] K. Hinsen, Analysis of domain motions by approximate normal mode calculations, Proteins: Structure, Function, and Bioinformatics 33 (3) (1998) 417–429.

[59] F. Tama, Y. Sanejouand, Conformational change of proteins arising from normal mode calculations, Protein Engineering 14 (1) (2001) 1–6.

[60] 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 70 (1) (2008) 131–143.

[61] I. Lotan, F. Schwarzer, Approximation of protein structure for fast similarity measures, Journal of Computational Biology 11 (2-3) (2004) 299–317.

[62] A. Shehu, B. Olson, Guiding the search for native-like protein conformations with an ab-initio tree-based exploration, International Journal of Robotics Research 29 (8) (2010) 1106–1127.

[63] J. Cortés, L. Jaillet, T. Siméon, Molecular disassembly with RRT-like algorithms, in: IEEE International Conference on Robotics and Automation, 2007, pp. 3301–3306.

[64] E. Plaku, H. Stamati, C. Clementi, L. Kavraki, Fast and reliable analysis of molecular motion using proximity relations and dimensionality reduction, Proteins: Structure, Function, and Bioinformatics 67 (4) (2007) 897–907.

[65] P. Jiménez, F. Thomas, C. Torras, 3d collision detection: a survey, Computers & Graphics 25 (2) (2001) 269–285.

[66] M. Lin, D. Manocha, Collision and proximity queries, in: Handbook of Discrete and Computational Geometry, 2003.

[67] 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, ACM, 1996, pp. 171–180.

[68] G. van den Bergen, Efficient collision detection of complex deformable models using aabb trees, Journal of Graphics Tools 2 (4) (1998) 1–13.

[69] J. Cohen, M. 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, ACM, 1995, pp. 189–196.

[70] M. Soss, J. Erickson, M. Overmars, Preprocessing chains for fast dihedral rotations is hard or even impossible, Computational Geometry 26 (3) (2003) 235–246.

[71] P. Agarwal, L. Guibas, A. Nguyen, D. Russel, L. Zhang, Collision detection for deforming necklaces, Computational Geometry 28 (2-3) (2004) 137–163.

[72] I. Lotan, F. Schwarzer, D. Halperin, J. Latombe, Efficient maintenance and self-collision testing for kinematic chains, in: Proceedings of the eighteenth annual symposium on Computational geometry, ACM, 2002, pp. 43–52.

[73] V. 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, MIT Press, 2005, pp. 241–248.

[74] H. Rangwala, G. Karypis, Protein Structure Methods and Algorithms, Vol. 14 of Wiley Series in Bioinformatics: Computational Techniques and Engineering, John Wiley & Sons, 2010.

[75] E. Coutsias, C. Seok, M. Jacobson, K. Dill, A kinematic view of loop closure, Journal of computational chemistry 25 (4) (2004) 510–528.

[76] R. Kolodny, L. Guibas, M. Levitt, P. Koehl, Inverse kinematics in biology: The protein loop closure problem, International Journal of Robotics Research 24 (2-3) (2005) 151–163.

[77] J. Cortés, T. Siméon, Sampling-based motion planning under kinematic loop-closure constraints, Algorithmic Foundations of Robotics VI (2005) 75–90.

[78] 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 21 (suppl 1) (2005) i116–i125.

[79] 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 5 (2008) 534–545.

[80] A. Canutescu, R. Dunbrack Jr, Cyclic coordinate descent: A robotics algorithm for protein loop closure, Protein Science 12 (5) (2003) 963–972.

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

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

[83] J. Ponder, D. Case, Force fields for protein simulations, Advances in protein chemistry 66 (2003) 27–85.

[84] A. Mackerell Jr, Empirical force fields for biological macromolecules: overview and issues, Journal of Computational Chemistry 25 (13) (2004) 1584–1604.

[85] V. Tozzini, Coarse-grained models for proteins, Current opinion in structural biology 15 (2) (2005) 144–150.

[86] L. Monticelli, S. Kandasamy, X. Periole, R. Larson, D. Tieleman, S. Marrink, The martini coarse-grained force field: extension to proteins, Journal of Chemical Theory and Computation 4 (5) (2008) 819–834.

[87] P. Derreumaux, From polypeptide sequences to structures using monte carlo simulations and an optimized potential, Journal of Chemical Physics 111 (5) (1999) 2301–2310.

[88] A. Singh, J. Latombe, D. Brutlag, A motion planning approach to flexible ligand binding, in: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, AAAI Press, 1999, pp. 252–261.

[89] M. Apaydin, A. Singh, D. Brutlag, J. Latombe, Capturing molecular energy landscapes with probabilistic conformational roadmaps, in: Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, 2001, pp. 932–939.

[90] M. Apaydin, C. Guestrin, C. Varma, D. Brutlag, J. Latombe, Stochastic roadmap simulation for the study of ligand-protein interactions, Bioinformatics 18 (Suppl 2) (2002) S18–S26.

[91] M. Apaydin, D. Brutlag, C. Guestrin, D. Hsu, J. Latombe, C. Varma, Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion, Journal of Computational Biology 10 (3-4) (2003) 257–281.

[92] M. Apaydin, D. Brutlag, D. Hsu, J. Latombe, Stochastic conformational roadmaps for computing ensemble properties of molecular motion, Algorithmic Foundations of Robotics V (2004) 131–147.

[93] T. Chiang, M. Apaydin, D. Brutlag, D. Hsu, J. Latombe, Predicting experimental quantities in protein folding kinetics using stochastic roadmap simulation, in: Research in Computational Molecular Biology, Springer, 2006, pp. 410–424.

[94] T. Chiang, M. Apaydin, D. Brutlag, D. Hsu, J. Latombe, Using stochastic roadmap simulation to predict experimental quantities in protein folding kinetics: folding rates and phi-values, Journal of Computational Biology 14 (5) (2007) 578–593.

[95] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, et al., Equation of state calculations by fast computing machines, Journal of Chemical Physics 21 (6) (1953) 1087.

[96] N. Amato, G. Song, Using motion planning to study protein folding pathways, Journal of Computational Biology 9 (2) (2002) 149–168.

[97] G. Song, S. Thomas, K. Dill, J. Scholtz, N. 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, 2003, pp. 240–251.

[98] N. Amato, K. Dill, G. Song, Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures, Journal of Computational Biology 10 (3-4) (2003) 239–255.

[99] S. Thomas, G. Song, N. Amato, Protein folding by motion planning, Physical biology 2 (2005) 148–155.

[100] X. Tang, B. Kirkpatrick, S. Thomas, G. Song, N. Amato, Using motion planning to study rna folding kinetics, Journal of Computational Biology 12 (6) (2005) 862–881.

[101] L. Tapia, X. Tang, S. Thomas, N. Amato, Kinetics analysis methods for approximate folding landscapes, Bioinformatics 23 (13) (2007) 539–548.

[102] X. Tang, S. Thomas, L. Tapia, D. Giedroc, N. Amato, Simulating rna folding kinetics on approximated energy landscapes, Journal of Molecular Biology 381 (4) (2008) 1055–1067.

[103] L. Tapia, S. Thomas, N. Amato, A motion planning approach to studying molecular motions, Communications in Information & Systems 10 (1) (2010) 53–68.

[104] 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 3 (1) (2007) 17–25.

[105] 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 94 (5) (2008) 1622–1629.

[106] J. Cortés, L. Jaillet, T. Siméon, Disassembly path planning for complex articulated objects, IEEE Transactions on Robotics 24 (2) (2008) 475–481.

[107] L. Jaillet, J. Cortés, T. Siméon, Sampling-based path planning on configuration-space costmaps, IEEE Transactions on Robotics 26 (4) (2010) 635–646.

[108] L. Jaillet, F. Corcho, J. Pérez, J. Cortés, Randomized tree construction algorithm to explore energy landscapes, Journal of Computational Chemistry.

[109] B. Raveh, A. Enosh, O. Schueler-Furman, D. Halperin, Rapid sampling of molecular motions with prior information constraints, PLoS Computational Biology 5 (2).

[110] 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 95 (8) (2008) 3850–3860.

[111] N. Haspel, M. Moll, M. Baker, W. Chiu, L. Kavraki, Tracing conformational changes in proteins, BMC Structural Biology 10 (Suppl 1) (2010) S1.

[112] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, science 220 (4598) (1983) 671–680.

[113] J. Cortés, T. Siméon, M. Remaud-Siméon, V. Tran, Geometric algorithms for the conformational analysis of long protein loops, Journal of Computational Chemistry 25 (7) (2004) 956–967.

[114] 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. In Press.

[115] C. Dobson, Protein folding and misfolding, Nature 426 (6968) (2003) 884–890.

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

[117] J. Balbach, V. Forge, N. van Nuland, S. Winder, P. Hore, C. Dobson, Following protein folding in real time using NMR spectroscopy, Nature Structural & Molecular Biology 2 (10) (1995) 865–870.

[118] H. Dyson, P. Wright, Unfolded proteins and protein folding studied by NMR, Chem. Rev 104 (8) (2004) 3607–3622.

[119] C. Chan, Y. Hu, S. Takahashi, D. Rousseau, W. Eaton, J. Hofrichter, Submillisecond protein folding kinetics studied by ultrarapid mixing, Proceedings of the National Academy of Sciences of the United States of America 94 (5) (1997) 1779–1784.

[120] C. Jones, E. Henry, Y. Hu, C. Chan, S. Luck, A. Bhuyan, H. Roder, J. Hofrichter, W. Eaton, Fast events in protein folding initiated by nanosecond laser photolysis, Proceedings of the National Academy of Sciences of the United States of America 90 (24) (1993) 11860–11864.

[121] R. Unger, J. Moult, Genetic algorithms for protein folding simulations, Journal of Molecular Biology 231 (1) (1993) 75–81.

[122] J. Onuchic, P. Wolynes, Theory of protein folding, Current Opinion in Structural Biology 14 (1) (2004) 70–75.

[123] K. Dill, S. Ozkan, M. Shell, T. Weikl, The protein folding problem, Annual review of biophysics 37 (2008) 289–316.

[124] J. Bryngelson, J. Onuchic, N. Socci, P. Wolynes, Funnels, pathways, and the energy landscape of protein folding: a synthesis, Proteins: Structure, Function, and Bioinformatics 21 (3) (1995) 167–195.

[125] D. Goodsell, G. Morris, A. Olson, Automated docking of flexible ligands: applications of autodock, Journal of Molecular Recognition 9 (1) (1996) 1–5.

[126] P. Lang, S. Brozell, S. Mukherjee, E. Pettersen, E. Meng, V. Thomas, R. Rizzo, D. Case, T. James, I. Kuntz, Dock 6: Combining techniques to model rna–small molecule complexes, RNA 15 (6) (2009) 1219–1230.

[127] M. Rarey, B. Kramer, T. Lengauer, G. Klebe, A fast flexible docking method using an incremental construction algorithm, Journal of Molecular Biology 261 (3) (1996) 470–489.

[128] G. Jones, P. Willett, R. Glen, A. Leach, R. Taylor, Development and validation of a genetic algorithm for flexible docking, Journal of Molecular Biology 267 (3) (1997) 727–748.

[129] 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 15 (5) (1994) 488–506.

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

[131] P. Hajduk, J. Greer, A decade of fragment-based drug design: strategic advances and lessons learned, Nature Reviews Drug Discovery 6 (3) (2007) 211–219.

[132] S. Sousa, P. Fernandes, M. Ramos, Protein–ligand docking: current status and future challenges, Proteins: Structure, Function, and Bioinformatics 65 (1) (2006) 15–26.

[133] 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 9 (8) (2008) 1308–1317.

[134] 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 10 (17) (2009) 2760–2771.