

Geometric Algorithms for the Conformational Analysis of Long Protein Loops

J. Cortés¹, T. Siméon¹, M. Remaud-Siméon², V. Tran³

¹ LAAS-CNRS, 7 avenue du Colonel-Roche, 31077 Toulouse, France

{jcortes,nic}@laas.fr

² Centre de Bioingenierie Gilbert Durand, 135 avenue de Ranguel, 31077 Toulouse, France

remaud@insa-tlse.fr

³ Unité de Recherches sur la Biocatalyse, 2 rue de la Houssinière, 44322 Nantes, France

Vinh.Tran@chimbio.univ-nantes.fr

Abstract

The efficient filtering of unfeasible conformations would considerably benefit the exploration of the conformational space when searching for minimum energy structures or during molecular simulation. The most important conditions for filtering are the maintenance of molecular chain integrity and the avoidance of steric clashes. These conditions can be seen as geometric constraints on a molecular model. In this paper, we discuss how techniques issued from recent research in Robotics can be applied to this filtering. Two complementary techniques are presented: one for conformational sampling and another for computing conformational changes satisfying such geometric constraints. The main interest of the proposed techniques is their application to the structural analysis of long protein loops. First experimental results demonstrate the efficacy of the approach for studying the mobility of loop 7 in amylosucrase from *Neisseria polysaccharea*. The supposed motions of this 17-residue loop would play a very important role in the activity of this enzyme.

Keywords : conformational search; robotic motion planning; loop closure; steric clash avoidance; long protein loops

1 Introduction

Prime techniques in structural investigations require the exploration of the conformational space \mathcal{C} of a molecule. Conformational search methods [1] explore \mathcal{C} in order to identify the stable structures of molecules, which determine their properties and functions. Molecular simulations [2] explore \mathcal{C} while computing conformational changes on a molecule under modified environmental conditions. The analysis of such changes of the molecular structure is essential for the understanding of many biological processes.

Since the goal of the conformational search is to find minimum energy structures, the exploration is much more efficient when it is limited to a subset of \mathcal{C} excluding energetically unacceptable conformations. Conformational changes explored in simulations can occur only if there is not a high energetic barrier to overcome. Therefore, approaches treating these problems will greatly benefit from efficient techniques able to provide samples and paths in \mathcal{C} that filter most unfeasible conformations.

The conformational analysis of a whole macromolecule is a very difficult problem. From a methodological point of view, two stages are usually necessary. The first stage corresponds to the identification of rigid segments (i.e. secondary structural elements) capable of participating in the molecular framework. The second stage is devoted to the remaining segments, so-called loops, assumed to be much more flexible. However, available techniques to predict low energy conformations of long loops are limited and much less efficient because of the loop flexibility.

When the global molecular architecture is assumed to be known and only portions (loops) are studied separately, the integrity of molecular chains must be maintained. The first and last atoms of the treated segment of a molecular chain must remain bonded with their neighbor atoms. Breaking these bonds requires a very high amount of energy. A strong constraint is thus imposed for the conformational exploration. This same constraint is present in the analysis of cyclic molecules. It is often referred to in the literature as the *loop-closure* constraint. Three main kinds of method can be applied to solve the loop-closing problem (i.e. computing conformations satisfying loop closure): analytical (e.g. [3, 4, 5]), optimization-based (e.g. [6, 7, 8]) and database methods

(e.g. [9, 10]). The difficulty of this problem increases with the length of the molecular chain, and available techniques are limited, or at least strongly penalized, by this.

In addition to breaking bonds, another large amount of energy is required to get two non-bonded atoms significantly closer than the sum of their Van der Waals (VdW) radii. A violation of this condition is called *steric clash*. Feasible conformations of a molecular segment cannot contain either internal clashes, which we call *self-clashes*, or clashes with atoms of the rest of the molecule. A possible filter for such unacceptable conformations consists of evaluating the repulsive term of the VdW energy and discarding conformations that exceed a given cutoff value [11]. However, this energetic constraint can also be treated by geometric procedures. The use of “clash grids”, computed from the distances between atoms, to perform this filtering was proposed in [12]. An interesting alternative is the use of collision detection algorithms applied on a three-dimensional (3D) model of the molecule [13]. Obviously, the higher the number of atoms, the more critical the efficiency of the technique.

In Robotics, the same kinds of constraint appear when treating the motion planning problem [14]. Paths must be computed in the subset of feasible configurations¹ of the robot, \mathcal{C}_{feas} . The main feasibility condition is collision avoidance. The robot cannot collide with obstacles in the workspace and self-collisions are also forbidden. Besides, when the robotic mechanism contains kinematic loops, closure constraints must be considered in the computed motions. Sampling-based motion planning techniques (e.g. [15, 16, 17]) have demonstrated to be efficient and general tools in this field. These techniques capture the topology of \mathcal{C}_{feas} within data structures (graphs or trees) by performing a random (or quasi-random) exploration of \mathcal{C} on a model of the robot and its environment.

In recent publications [18, 19], we have described efficient algorithms for planning motions of closed-chain mechanisms. In this paper, we investigate the adaptation of these techniques to handle molecular models. Although the method could be applied to any molecular segment or cyclic molecule, we are mainly interested in the application to long protein loops.

2 Interest in Protein Loops

Loops play key roles in the function of proteins. They are often involved in active and binding sites. Therefore, when predicting a protein structure, an accurate loop modeling is necessary for determining its functional specificity.

Modeling loops in proteins is one of the main open problems in Structural Biology. Comparative-modeling methods (see [20] for a survey) often fail in the prediction of protein loop structures when the percentage of sequence identities between known and predicted protein family members is low. Indeed, it is well established that there is no reliable approach for modeling long loops (more than five residues) available at this time [21].

The alternatives to comparative-modeling are *de novo* (or *ab initio*) methods [22]. Such methods carry out a search of low energy conformations for a given amino acid sequence. Many different approaches have been proposed for modeling protein loops. One of the most developed techniques is described in [23]. This reference paper also provides a concise survey of loop modeling methods. The accuracy of *de novo* methods mainly depends on the energy function they use. Therefore, improvements in the results provided by these approaches require the design of fine energy models. However, progress in the conformational exploration strategies may also be necessary in order to increase the efficiency of these techniques which are today computationally expensive.

Even more important than the prediction of stable loop conformations is the determination of the feasible conformational changes. In many enzymes, for example, surface loops undergo conformational changes to catalyze a reaction [24]. Furthermore, loop motions are generally involved in protein interactions. Therefore, introducing loop flexibility into docking approaches is necessary for a more accurate prediction of these interactions [25].

3 Aim of Our Approach

The techniques proposed in this paper aim to be new tools for the structural analysis of long polypeptide segments, and, in particular, of protein loops. The efficiency of geometric algorithms

developed in the field of Robotics can relieve conformational exploration approaches of a part of the heavy energetic treatment.

In Section 5, we propose a conformational sampling technique that generates random conformations satisfying loop-closure and clash avoidance constraints. The backbone conformation is first computed by an algorithm that relies on efficient geometric and kinematic procedures. Side-chain conformations are then generated by combining sampling techniques and an effective collision detection algorithm. Families of approaches requiring conformational sampling, such as Monte Carlo algorithms [26] or Stochastic Roadmap techniques [27], would directly benefit from such filtered conformations.

Another interesting feature of our sampling technique is to compute loop conformations avoiding steric clashes with the rest of the protein. Using this technique to compute random samples uniformly distributed in the conformational space will provide very useful information about the allowed conformations of the loop in its environment. For instance, this information could be represented in the form of Ramachandran plots [28], and techniques (e.g. MODELLER [23]) using such statistical distributions could gain in performance.

The geometric analysis can be pushed further. In Section 6, we propose an algorithm to capture the connectivity of the sub-space of geometrically feasible conformations. The possible deformations maintaining loop-closure and clash avoidance constraints are explored and encoded in a data structure. Such data structure would be useful for many existing conformational exploration approaches. Note that a conformational search method sharing similar ideas has been proposed in [29] for small molecules (ligands) under geometric constraints.

4 Problem Formulation

The problem is formulated from a robotic point of view. First, the geometric model of the molecule is described. The constraints that must be satisfied during the exploration of the conformational space are then defined.

4.1 Geometric Model

4.1.1 Kinematics Inspired Model

A molecule is a set of atoms \mathcal{A}_i partially connected by bonds. A sequence of bonded atoms is called a *molecular chain*. Three parameters, usually called *internal coordinates*, define the relative position of consecutive atoms in a molecular chain: bond lengths, bond angles, and dihedral angles. The widely adopted *rigid geometry assumption* (see [30] as one of the first references) considers that only dihedral angles are variable parameters. Under this assumption, a molecule can be seen as an articulated mechanism with *revolute joints* between bonded atoms. The model of a molecular chain can be built from the internal coordinates using kinematics conventions. We follow the *modified-Denavit-Hartenberg* (mDH) convention described in [31]. A Cartesian coordinate system F_i is attached to each atom \mathcal{A}_i and then the relative location of consecutive frames can be defined by a homogeneous transformation matrix:

$${}^{i-1}T_i = \begin{pmatrix} \mathbf{C}\theta_i & -\mathbf{S}\theta_i & 0 & 0 \\ \mathbf{S}\theta_i\mathbf{C}\alpha_{i-1} & \mathbf{C}\theta_i\mathbf{C}\alpha_{i-1} & -\mathbf{S}\alpha_{i-1} & -\mathbf{S}\alpha_{i-1}d_i \\ \mathbf{S}\theta_i\mathbf{S}\alpha_{i-1} & \mathbf{C}\theta_i\mathbf{S}\alpha_{i-1} & \mathbf{C}\alpha_{i-1} & \mathbf{C}\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where d_i is the bond length between atoms \mathcal{A}_{i-1} and \mathcal{A}_i ; α_{i-1} is the supplement of the bond angle between \mathcal{A}_{i-2} , \mathcal{A}_{i-1} and \mathcal{A}_i ; θ_i is the dihedral angle formed by atoms \mathcal{A}_{i-2} , \mathcal{A}_{i-1} , \mathcal{A}_i and \mathcal{A}_{i+1} (see Figure 1.a). \mathbf{C} and \mathbf{S} represent sines and cosines respectively.

A molecular chain between atoms \mathcal{A}_0 and \mathcal{A}_n is then modeled by a kinematic chain, ${}^1\mathcal{K}_n$, in which joint variables correspond to dihedral angles. The conformation of the chain is determined by the array q of the θ_i . The kinematic model of a polypeptide segment is composed of a set of chains: the main-chain (the backbone) and the side-chains, which are built upon it. The conformation of the segment is then specified by an array containing the conformation parameters of the backbone and of all the side-chains.

Often, some portions of molecular models are treated as rigid solids, for instance, peptide units in proteins. The rigid geometry assumption also considers that double bond torsion angles, such as peptide bonds, are fixed. Hence, the number of frames required in the kinematic modeling is reduced. Figure 1.b illustrates how the frames corresponding to the mDH parameters are obtained by simple geometrical operations when the dihedral angle associated with a peptide bond ω is fixed at a given value. Thus, several atoms in each peptide unit have constant coordinates in these frames. As proposed in a recent work [32], frames only need to be attached to rigid units (called *atomgroups* by the authors). Then, the relative location of atoms in an atomgroup only requires positional coordinates, yielding to a more efficient method for updating conformations.

4.1.2 Van der Waals Model

The VdW model consists of a representation of the molecule by the union of solid spheres associated with atoms. A VdW radius is assigned to each atom type. This geometric model of the molecule is the simplest and most ordinary space filling diagram [33]. In molecular models treated by our approach, such spheres are the mobile *bodies* of the articulated polypeptide segment and the static *obstacles* corresponding to the rest of the atoms in the molecule, which compose what we call the *environment*.

4.2 Geometric Constraints

4.2.1 Loop Closure

A loop-closure constraint applied on the kinematic model of a molecular chain ${}^1\mathcal{K}_n$ fixes the relative location of the frames F_0 and F_n , which we call *base-frame* and *end-frame* respectively. Therefore, the transform matrix 0T_n is known. This matrix can also be obtained from the sequence of local transformations:

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

This equality provides a system of equations, called *closure equations*, where the unknowns are the joint variables θ_i . Hence, a relationship must exist between the parameters in q for satisfying loop closure.

4.2.2 Clash Avoidance

Distances between non-bonded atoms that are substantially shorter than the sum of their VdW radii must be avoided. The choice of the limiting contact distance is ambiguous. For our experiments, we model molecules using a percentage (usually 70%) of the VdW radii proposed in [34]. Collisions between such reduced VdW spheres must be avoided if they are separated by more than three bonds. This condition must be satisfied between the atoms of the articulated segment and between these atoms and the static atoms of the rest of the molecule.

5 Conformational Sampling

Algorithm 1 computes a random conformation of a polypeptide segment (the protein loop) achieving loop-closure and clash avoidance constraints on the 3D model. First, the backbone conformation q_b is generated. The procedure for obtaining random conformations satisfying closure is explained in Section 5.1. These conformations are then tested for clashes of backbone atoms between themselves and with atoms in the environment. Once a feasible conformation for the backbone has been computed, random conformations of the side-chains q_s are tested. These chains are built iteratively until all of them are free of clashes. Section 5.2 explains the process.

5.1 Backbone Conformation with Closure

Obtaining a backbone conformation satisfying loop closure requires the solution of the closure equations mentioned in Section 4.2.1. Unfortunately, and despite the intensive research in the field, no efficient general solution is currently available to solve systems of multi-variable non-linear algebraic equations (see [35] for a survey).

It is now well known that, in general, six variables in the closure equations are dependent

on the rest (independent variables). Notice that six is the minimum number of parameters that allow to span full-rank subsets of $SE(3)$ (the position-orientation space in a 3D world) [14]. Many articles in Computational Chemistry and Robotics (e.g. [3, 5, 36, 37, 38, 39]) propose methods to obtain these six dependent variables as a function of the other parameters. Except for very particular geometries (e.g. regular cyclohexane [40]), only a finite number of solutions exists.

The remaining difficulty is how to obtain values for the independent variables for which a solution of the closure equations exists. In Robotics, detailed analytical approaches have been proposed only for planar or spherical closed mechanisms [41]. In Computational Chemistry, only a few authors have tackled this problem. Decimation approaches and hierarchical decomposition of the closing problem have been proposed for loops with six or more residues [5]. However, for very long loops, the efficiency of such methods decreases because closure equations must be solved several times for different fragments of the chain.

We propose an algorithm, called *Random Loop Generator* (RLG), which produces random configurations of articulated mechanisms containing closed chains. This algorithm has demonstrated its efficiency within robotic motion planning techniques [18, 19]. The configuration parameters of a closed kinematic chain are separated into two arrays: we call the independent variables of the closure equations the *active variables* q^a , and the dependent variables the *passive variables* q^p . The RLG algorithm performs a particular random sampling for q^a that notably increases the probability of obtaining solutions for q^p .

We next explain the main elements of our approach and how it can be applied to polypeptide backbone segments. Explanations are illustrated on a very simple mechanism, the $6R$ planar linkage in Figure 2. The \mathcal{L}_i are the rigid bodies and the J_i are the revolute joints connecting them.

5.1.1 Loop Decomposition

The choice of the dependent and independent variables in the closure equations is arbitrary. We choose them consecutively in the kinematic chain. Thus, we can refer to a *passive sub-chain*

involving joints whose variables are in q^p (*passive joints*). Although the passive sub-chain can be placed anywhere in the closed chain, it is convenient to place it in the middle. In the general case, the passive sub-chain is a mechanism with six degrees of freedom. For a polypeptide backbone model under the rigid geometry assumption, only dihedral angles ϕ and ψ are variable. Therefore, the passive sub-chain is composed of the backbone of three residues. In the example in Figure 2, three consecutive revolute joints (i.e. a $3R$ planar mechanism) are sufficient. We have chosen J_3 , J_4 and J_5 to be the passive joints of the $6R$ linkage. Then, the rest of the joints (*active joints* corresponding to q^a) can be seen as contained in two *active sub-chains* rooted on the (fictive) solid on which the base-frame and the end-frame are fixed ($\mathcal{L}_{0,6}$ in Figure 2).

5.1.2 RLG Algorithm

The pseudo-code of the algorithm that generates random configurations of a single-loop closed chain (applied to the loop backbone) is synthesized in Algorithm 2. First, the configuration parameters of the active sub-chains, q^a , are computed by the function `SAMPLE- q^a` detailed in Algorithm 3. The idea of the algorithm is to progressively decrease the complexity of the closed chain treated at each iteration until only the configuration of the passive sub-chain, q^p , remains to be solved. The two active sub-chains are treated alternately. The ideal solution should be to sample each joint variable from the subset of values, which we call *closure range*, satisfying the closure equations. However, computing this subset is as difficult as solving the general closure equations. Thus, an approximation is used. This approximation must be conservative in order to guarantee a complete solution (i.e. no region of the sub-space satisfying closure constraints is excluded from the sampling). More details about how to obtain the approximated closure range are given in Section 5.1.3. The closure range of the joint variable treated at one iteration depends on the configuration of the previously treated joints. Hence, this subset must be recomputed for all the joints (except the first treated one) in the generation of each new configuration. Because of the conservative nature of the approach, it is possible to obtain an empty set. In this case, the process is restarted.

Figures 2.a,b,c illustrate how the values of θ_1 , θ_6 and θ_2 (the active variables q^a) are generated for the $6R$ linkage. For each joint variable, the estimation of the closure range is computed and a random value is sampled inside this set. Figure 2.d shows the two solutions of the closure equations for the passive sub-chain. In this case, these solutions are obtained by simple trigonometric operations. The solution for the passive sub-chain in the polypeptide backbone model is treated in Section 5.1.4.

5.1.3 Computing Closure Range

The problem can be formulated as follows. Given a closed kinematic chain ${}^b\mathcal{K}_e$ involving joints from J_b to J_e (we consider $b < e$ in this explanation), two open kinematic chains are obtained by breaking the body \mathcal{L}_b between J_b and J_{b+1} . A suitable break-point is the physical placement of J_{b+1} , but any other point can be chosen. A frame F_C associated with this break-point can be seen as the end-frame of both open chains. The closure range of the joint variable corresponding to J_b , θ_b , is the subset of values for which F_C is reachable by the open chain ${}^e\mathcal{K}_{b+1}$. In general, the exact solution to this problem is extremely complex. Most works in the Robot Kinematics literature are limited to particular instances (e.g. [42, 43]). For our purpose, a simple and fast method is preferred to a more accurate but slower one. We solve the problem only considering positional reachability.

Since J_b is a revolute joint, the origin of F_C describes a circle around its axis. The approximation of the closure range is obtained by the intersection of this circle with a volume (surface for the planar case in Figure 2) bounding the region mapped by the origin of F_C attached to the chain ${}^e\mathcal{K}_{b+1}$, which is called the *reachable workspace* (RWS) in Robotics. This bounding volume is contained between two concentric spheres (circles) centered at the origin of the base-frame and whose radii are the maximum and minimum extension of the chain, r_{ext} and r_{int} . For a general mechanism, obtaining these radii requires the solution of complex optimization problems. If the appropriate (even if computationally slow) method is available, it can be used in a pre-computing phase. However, simpler particular solutions can be adopted for particular classes of mechanisms.

The solution is straightforward for the planar linkage in our example. The regions designated as RWS in Figures 2.a,b,c represent such bounding surfaces at different steps of the algorithm.

In the application to molecular models, frames F_C are the frames attached to atoms. Particularities in the geometry of polypeptide backbones allow the design of a simple approximated method to compute the spheres bounding RWS. For chains containing more than three residues (which is the size of the passive sub-chain), r_{int} can be simply considered zero without decreasing the performance of the technique.

The maximum distance between the extreme atoms of a segment of polypeptide backbone² is often obtained for a conformation with all the dihedral angles at π . We call this length l_π . However, this assumption is not always true, in particular if a slight rotation around peptide bonds is allowed. An upper bound of the maximum is required for guaranteeing completeness. This upper bound \hat{l} is the sum of the distances between consecutive C_α atoms (i.e. the length of peptide units). Obviously, when the chain begins or ends with a fragment of a peptide unit (i.e. only one or two of the three concerned atoms in the backbone are contained in the chain), the length of this portion is added. Instead of using a constant value, r_{ext} is sampled from a distribution between l_π and \hat{l} each time this dimension is required in the process. We suggest using a Gaussian distribution with $\mu = l_\pi$ and $\sigma^2 = 1$. This increases the efficiency of the approach while keeping completeness.

This approximated method to obtain r_{ext} is not dependent on a particular kind of geometry. It can be applied on standard models or to structures acquired from the Protein Data Bank (PDB) (<http://www.rcsb.org/pdb>). Figure 3 illustrates the application to backbone segments with standard Pauling-Corey geometry [44].

5.1.4 General 6R Inverse Kinematics

The kinematic model of the three-residue backbone corresponding to the passive sub-chain in our approach can be seen as a 6R manipulator with general geometry [45]. Obtaining the conformation of a serial manipulator, given the location of the base-frame and the end-frame (i.e. solving the

closure equations), is known in Robotics as the *inverse kinematics problem*.

The method we use to solve the general $6R$ inverse kinematics problem is inspired by the work of Lee and Liang [36]. The principle of the method is described in [39]³. The algebraic elimination of variables starts in a way similar to that used in related works (e.g. [37, 38]). However, Renaud goes further in the elimination process, arriving at an 8×8 quadratic polynomial matrix in one variable instead of the 12×12 matrix in the referred methods. The problem can then be treated as a generalized eigenvalue problem (as previously proposed in [38]), for which efficient and robust solutions are available [46]. Another important advantage of the method in relation to all previous approaches is that it requires a minimum number of divisions in the elimination process. In particular, divisions by zero are avoided in order to guarantee robustness.

Note that the general $6R$ mechanism can have up to 16 inverse kinematic solutions. Thus, several sets of values of the passive variables q^p satisfy loop closure equations for a given value of the active variables q^a . Each backbone conformation obtained by composing q^a with the different q^p is treated by the algorithm RANDOMLOOPCONF (Algorithm 1).

5.2 Clashes and Side-Chain Conformation

5.2.1 Collision Detection Algorithms

A collision detection algorithm determines if contacts or penetrations exist between 3D bodies. They are important tools in Computational Geometry and Robotics [47, 48]. Collision detection is the most computationally expensive process in sampling-based motion planning techniques. Thus, effective algorithms have been developed in this field to try to minimize this cost.

In our current implementation of the approach, clashes in a sampled conformation are checked by a generic collision detection algorithm [49], which operates well within geometrically complex 3D scenes.

5.2.2 Sampling Side-Chain Conformation

The conformation of the side-chains is built upon a feasible backbone conformation. These side-chain conformations are generated by randomly sampling the side-chain dihedral angles and tested until a collision-free solution is found. A progressive construction is carried out. Instead of rebuilding all the side-chains when the collision test is positive, only the conformation of clashing side-chains is resampled. The resampling and collision detection process is performed following an arbitrary order of the side-chains, intending to prevent a privileged conformational sampling. When two side-chains collide together, but self-clashes or clashes with the backbone and the rest of the protein do not exist, only one of them will be resampled. The process is iterated a certain number of times before returning that a clash-free conformation of the side-chains cannot be found.

6 Conformational Space Exploration

6.1 Sampling-based Motion Planning Techniques

Sampling-based motion planning techniques appeared in Robotics as an alternative to exact approaches [14] that cannot be applied to high-dimensional configuration-spaces. In particular, algorithms based on the *Probabilistic RoadMap* (PRM) approach (e.g. [15, 16]) have mostly been developed. The general PRM principle is to construct a graph (roadmap) that captures the topology of the feasible subset of robot configurations, \mathcal{C}_{feas} . The nodes of this graph are randomly sampled configurations satisfying intrinsic conditions in this subset (e.g. collision avoidance). The edges are short feasible paths (*local paths*) linking “nearby” nodes. Other families of methods aim to efficiently solve *single planning queries* instead of covering the whole search-space. The *Rapidly-exploring Random Tree* (RRT) [17] is a data structure and sampling scheme to quickly search high-dimensional constrained spaces. \mathcal{C}_{feas} is explored by one or two trees rooted at the start and/or goal configurations. The exploration is biased by sampling points in \mathcal{C} and incrementally pulling the search tree(s) toward them.

This section treats the application of these techniques onto geometric models of molecules.

Let us call \mathcal{C}_{clos} the subset of the conformations satisfying the loop-closure constraints and \mathcal{C}_{free} the subset of clash-free conformations. $\mathcal{C}_{feas} = \mathcal{C}_{clos} \cap \mathcal{C}_{free}$ is the subset of the geometrically feasible conformations to be explored. Obviously, not every conformation in \mathcal{C}_{feas} is energetically acceptable, but a significant number of high-energy structures are excluded from this subset. We assume that \mathcal{C}_{feas} contains all the energetically feasible conformations: $\mathcal{C}_{lowE} \subset \mathcal{C}_{feas} \subset \mathcal{C}$.

6.2 Incremental Search Keeping Constraints

We next explain an algorithm to carry out the incremental search of \mathcal{C}_{feas} using an RRT-like technique [17], extended to handle the geometric constraints in our problem. Algorithm 4 gives the pseudo-code and Figure 4 illustrates the exploration in a simple two-dimensional example. The darker regions in the figure correspond to conformations with steric clashes, \mathcal{C}_{free} being the rest of the space. In general, conformations satisfying closure (in \mathcal{C}_{clos}) are grouped into different disjoint continuous manifolds [50]. We have considered two manifolds \mathcal{C}_{clos}^1 and \mathcal{C}_{clos}^2 for this illustration.

The starting point q_{init} can be a randomly sampled feasible conformation (e.g. generated by the technique explained in Section 5) or a known conformation (e.g. acquired from the PDB). For executing an expansion step of the RRT, a random conformation q_{rand} is first sampled in \mathcal{C} . q_{rand} needs not to satisfy either closure or clash avoidance constraints. This conformation is only used as a local goal for the exploration. Nevertheless, we have experimentally shown that a *guided-random sampling* generating q_{rand} close to the subset satisfying closure equations improves the process (i.e. a wider portion of the space is explored in less time) in relation to a uniform random sampling [51]. For this, the configuration parameters corresponding to the independent variables of the closure equations, q^a , are generated by the function `SAMPLE_` q^a (Algorithm 3), explained in Section 5.1.2. Then, the nearest node in the current tree, q_{near} , is selected using a distance metric in \mathcal{C} . A new conformation q_{feas} is iteratively pulled from q_{near} toward q_{rand} . The pulled conformation must remain in the feasible subset. The closure constraint is maintained as follows. A conformation q_{step} is obtained by interpolating q_{near} and q_{rand} following a law (e.g. linear interpolation). The closure equations are then solved for the passive variables of the

backbone conformation (called q^p in Section 5.1). If the solution in the same manifold as q_{near} exists, then the conformation satisfying closure, q'_{step} , is checked for clashes. The process goes on until one of the feasibility conditions is violated. The new node of the tree, q_{new} , is an intermediate conformation between q_{near} and the last obtained q_{feas} . We use a Gaussian sampling between q_{feas} and q_{near} to obtain it.

Several criteria can be adopted for stopping the exploration. The simplest one is to build the tree until it contains a given number of nodes. The drawback is that this criterion is not related to a coverage of the explored region. We believe that an estimation of this coverage could be deduced from the number n_{fail} of consecutive times the algorithm fails when trying to expand the tree. A similar relationship has been demonstrated in related methods [16].

While the infinite solutions of the global inverse kinematics problem are grouped into different disjoint continuous manifolds and collision-free portions of each manifold can be also disjoint, the explained algorithm can explore only a region in \mathcal{C}_{feas} . Several starting points are required for exploring the different connected components of \mathcal{C}_{feas} . An algorithm combining RRT and PRM techniques could be used for the exploration of the whole subset.

6.3 Exploration with Flexible Geometry

Considering fixed values for bond lengths, bond angles and double bond torsion angles is a well accepted assumption that reduces the complexity of the structural analysis of molecules. However, it implies a severe restriction for conformational space exploration [52].

The rigid geometry assumption can be relaxed by allowing a slight variation of these parameters within given intervals. Handling these new variables is not a hard problem for our exploration algorithm, proceeding as follows. To generate a conformation q_{rand} , parameters d , α and ω (see Section 4.1.1) are first randomly sampled within the defined intervals. Then, the approach explained in Section 5 can be used. In the incremental variation of the selected conformation q_{near} toward q_{rand} , the new parameters are treated like the rest of the (non-passive) variables (i.e. they are interpolated following a given law).

7 First Results: Loop 7 Motions of Amylosucrase from *Neisseria Polysaccharea*

Amylosucrase (AS) is a glucansucrase that catalyzes the synthesis of an amylose-like polymer from sucrose. In the Carbohydrate-Active enZYme database (CAZy) (<http://afmb.cnrs-mrs.fr/~cazy/CAZY/index.htm>), this enzyme is classified in family 13 of glucoside-hydrolases (GH), which mainly contains starch converting enzymes (hydrolases or transglycosidases). Remarkably, this enzyme is the only polymerase acting on sucrose substrate reported in this family, all the other glucansucrases being gathered in GH family 70. Which structural features are involved in AS specificity is an important fundamental question. Indeed, the structural similarity of AS to family 13 enzymes is high. The 3D structure reveals an organization in 5 domains [53]. Three of them are commonly found in family 13: a catalytic $(\beta/\alpha)_8$ barrel domain, a B domain between β -strand 3 and α -helix 3 (loop 3) and a C terminal Greek key domain. Two additional domains are found in AS only: a helical N-terminal domain and a domain termed B', formed by an extended loop between β -strand 7 and α -helix 7. Domain B' partially covers the active site located at the bottom of a pocket and is mainly responsible for this typical architecture. Recently, co-crystallization of AS with maltoheptaose revealed the presence of two maltoheptaose binding sites, the first one (OB1) in the main access channel to the active site and a second one (OB2) at the surface of domain B'. Soaking AS crystals with sucrose also revealed the presence of a second sucrose binding site (SB2) different from the active site initially identified [54]. The comparison of the various structures obtained suggests that motion of the 17-residue fragment of domain B' starting at residue Gly⁴³³ and ending at residue Gly⁴⁴⁹, consecutive to oligosaccharide binding, could facilitate sucrose translocation from SB2 to the active site. In the following part, this fragment will be called loop 7. This loop could play a pivotal role responsible for the structural change and the polymerase activity. In this context, molecular simulation of loop 7 motion appears to be crucial to gain new insight into AS structure-function relationships.

Figure 5 shows the crystallographic structure of AS and the location of the residues we mention

in the following paragraphs. The model for our tests was created from the PDB file containing this structure (PDB ID: 1G5A), considering loop 7 as an articulated mechanism and the rest of the atoms as static elements. Atoms were modeled with 70% of their VdW radii. Images on the left in Figure 6 represent the articulated VdW model of the loop and a portion of its environment. Under our modeling assumptions, the results of the geometric exploration showed that only slight conformational variations of the loop are possible if the backbone integrity is maintained and steric clashes are avoided. The image on the right in Figure 6.a shows the skeleton of the articulated segment and a representation of one of the RRTs computed for this test. Nodes of the RRT are graphically represented by the positions explored by the C_α atom of Ser⁴⁴¹, the middle residue of the loop. This result contradicts pre-supposed significant loop fluctuations. Of course, our approach is not deterministic and therefore we cannot guarantee that such a motion does not exist. However, after several exhaustive tests, we can assert that the probability of its existence is very low. The average size of the constructed RRTs is 1000 nodes, for which about 4000 random conformations and 20000 complete collision tests were necessary. The average computing time was 1 hour⁴. It should be noted that computing time is mostly spent in collision detection. The generation of random conformations is very fast. For this loop, computing a conformation satisfying closure (including the update of all the frames and atom positions) takes less than 0.1 seconds with a non-optimized implementation. The conformational sampling used by the exploration algorithm (i.e. guided-random sampling of q^a without solving the closure equations for q^p) demands only about 0.01 seconds per conformation.

Several structural elements, and mainly loop 3 (residues 183-262), restrain the mobility of loop 7. Residue Asp²³¹ was identified as the main “geometric lock” responsible for the loop 7 enclosing. The side-chain of this residue was removed from the model in order to simulate a possible conformational change of this chain or even of the whole loop 3. The conformational exploration in this case showed that the loop is able to effect the expected motions keeping geometric constraints. The C_α atom of Ser⁴⁴¹ can be dislocated more than 9Å from its crystallographic position. Several tests were performed in order to see if the random nature of the approach could have an important

influence on the nature of the results. Similar motions were obtained for all of them. The loop moves almost as a rigid body with hinges at the extreme residues. Considerable variations of the backbone dihedral angles are concentrated in residues 433-436 and 446-449. Figure 6.b shows the representation of the RRT constructed in one of these tests. The images in Figure 7 correspond to four frames of the conformational change encoded in the RRT. Therefore, an “opening/closing” mechanism similar to other enzymes (e.g. [55, 24]), termed *conformational gating*, is suspected for this loop. The role that residue Asp²³¹ could play in this mechanism is being investigated. Directed mutagenesis experiments, replacing residue Asp²³¹ by glycine, are currently being developed.

8 Discussion and Prospects

We have proposed geometric techniques aimed at providing powerful filters for conformational sampling and search methods. Our solution to the loop-closing problem is computationally efficient and its performance is only slightly affected by the length of the molecular chain. To the best of our knowledge, only the CCD algorithm recently proposed in [8], offers a similar performance. While this optimization-based algorithm converges to an approximate closure solution starting from a nearly open conformation of the loop, our RLG sampling method computes exact solutions to the closure problem. Also, one disadvantage pointed out in [8] is that the CCD optimization technique which considers one degree of freedom at a time, may favor large changes in the first residues of the loop. By comparison, the random strategy of RLG produces more uniformly distributed samples, better suited for the exploration of the conformational space.

Our algorithms are currently implemented within the motion planning platform Move3D [56] developed at LAAS for robotics applications. No particular consideration has been given to reducing computation time in the present implementation which is aimed at demonstrating the efficacy of the proposed techniques. More extensive experimental tests and performance comparisons remain for future work, based on an optimized version taking advantage of the specifics of molecular models. We started the development of such an optimized version and a stand alone library that

could be made accessible to the scientific community.

Concerning the avoidance of steric clashes, collision detection algorithms combined with smart sampling techniques constitute an attractive alternative to methods producing optimization-based rearrangements. We are developing a tailored collision detection algorithm for molecular models which should perform faster than the generic checker currently used. In addition, a different progressive process for building backbone conformations is going to be tried. In contrast to the described sampling approach, clashes between the backbone atoms and the static environment will be checked after each step of the RLG algorithm.

In our current implementation, values for all variable dihedral angles in the side-chains and backbone are randomly sampled in the interval $(-\pi, \pi]$. As in other related techniques, our approach could handle information of the statistically preferred values of these angles (e.g. from Ramachandran plots by residue type). Using this information, many local steric clashes should be implicitly avoided.

Concerning the exploration technique, we are working on a method for pruning branches of the RRT in order to decrease the size of this data structure thus increasing the speed of the search process. Preliminary results using a visibility-based heuristic [16] seem promising.

The algorithms we have presented in this paper treat conformations of a molecular segment in a static environment. The extension of these algorithms to handle the flexibility of side-chains in this environment could be done without difficulty. Handling several loops which share the same region of the space (e.g. antibody hypervariable loops [57]) is an interesting extension we expect to develop.

The first results of the application of our robotic approach to molecular models show the potential of this technique. A fast geometric analysis can help to find the answer to important biochemical questions such as: what are the crucial residues in the biochemical reaction ? and what are the possible conformational changes ?

Although our next goal is to improve this geometrically constrained exploration, the final aim is to incorporate the energetic analysis into the incremental search technique. An energy function

can easily be integrated into this kind of exploration algorithm. Indeed, impressive results have been obtained by conformational search methods inspired by sampling-based motion planning techniques applied to computer assisted drug design [29], protein folding [27, 58] and ligand-protein docking [59, 60]. Given this energy function, geometrically feasible conformations generated by our approach could be evaluated and labeled, and then only the subset of the conformational space \mathcal{C}_{lowE} below a certain energetic limit should be explored.

Acknowledgments

The collaboration and the advice of the following people has been essential for carrying out this work: Gwénaëlle André (INRA, Nantes), Paul Bates (CRUK, London), Patrick Danes (LAAS-CNRS, Toulouse), David Guieysse (INSA, Toulouse), Marc Renaud (LAAS-CNRS, Toulouse), Lluís Ros (IRI-CSIC, Barcelona), Vicente Ruiz (IRI-CSIC, Barcelona). This work has been partially supported by the interdisciplinary CNRS project BioMove3D and the European project IST-37185 MOVIE.

Footnotes

¹ A configuration for a robot is the equivalent to a conformation for a molecule. We designate both, the configuration-space and the conformational space, by \mathcal{C} .

² Without proline. This case, studied apart, is not detailed in this paper.

³ The author is currently working on an extended version with full technique details.

⁴ Tests were performed using a Sun Blade 100 Workstation with a 500-MHz UltraSPARC-IIe processor.

References

- [1] Leach, A.R. *Molecular Modeling: Principles and Applications*; Longman, 1996; Chapter 8.
- [2] Frenkel, D.; Smit B. *Understanding Molecular Simulation: From Algorithms to Applications*; Academic Press, 1996.
- [3] Gō, N.; Scheraga, H.A. *Macromolecules* 1970, 3, 178-187.
- [4] Manocha D.; Zhu, Y.; Wright, W. *CABIOS* 1995, 11(1), 71-86.
- [5] Wedemeyer, W.J.; Scheraga, H.A. *J Comp Chem* 1999, 20, 819-844.
- [6] Shenkin, P.S.; Yarmush, D.L.; Fine R.M.; Wang, C.; Levinthal, C. *Biopolymers* 1987, 26, 2053-2085.
- [7] Zheng, Q.; Rosenfeld, R.; Vajda, S.; DeLisi, C. *J Comp Chem* 1993, 14, 556-565.
- [8] Canutescu, A.A.; Dunbrack Jr., R.L. *Protein Sci* 2003, 12, 963-972.
- [9] Oliva, B.; Bates, P.A.; Querol, E.; Aviles, F.X.; Sternberg, M.J.E *J Mol Biol* 1997, 266(4), 814-830.
- [10] van Vlijmen, H.W.T.; Karplus, M. *J Mol Biol* 1997, 267(4), 975-1001.
- [11] Bruccoleri, R.E.; Karplus, M. *Biopolymers* 1987, 26, 137-168.
- [12] Moult, J.; James, M.N.G. *Proteins* 1986, 1, 146-163.
- [13] Lotan, I.; Schwarzer, F.; Halperin, D.; Latombe, J.C. *Proc 18th ACM Symp Comp Geom* 2002, 43-52.
- [14] Latombe, J.C. *Robot Motion Planning*; Kluwer Academic Publishers, 1991.
- [15] Kavragi, L.E.; Švestka, P.; Latombe, J.C.; Overmars, M. *IEEE Trans Rob & Autom* 1996, 12(4), 566-580.
- [16] Siméon, T.; Laumond, J.P.; Nissoux, C. *Adv Rob J* 2000, 14(6), 477-494.

- [17] LaValle, S.M. in: Control Problems in Robotics; Springer, 2002; pp. 19-37.
- [18] Cortés, J.; Siméon, T.; Laumond, J.P. Proc IEEE Int Conf Rob & Autom 2002, 2141-2146.
- [19] Cortés, J.; Siméon, T. Proc IEEE Int Conf Rob & Autom 2003, 4354-4359.
- [20] Contreras-Moreira, B.; Fitzjohn, P.W.; Bates, P.A. App Bioinfo 2002, 1(4), 177-190.
- [21] Tramontano, A.; Leplae, R.; Morea, V. Proteins 2001, Suppl.5, 22-38.
- [22] Baker, D.; Šali, A. Science 2001, 295, 93-96.
- [23] Fiser, A.; Do, R.K.; Šali, A. Protein Sci 2000, 9, 1753-1773.
- [24] Osborne, M.J.; Schnell, J.; Benkovic, S.J.; Dyson, H.J.; Wright, P.E. Biochemistry 2001, 40(33), 9846-9859.
- [25] Janin, J.; Henrick, K.; Moult, J.; Eyck, L.T.; Sternberg, M.J.E.; Vajda, S.; Vakser, I.; Wodak, S.J. Proteins 2003, 52(1), 2-9.
- [26] Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. J Chem Phys 1953, 21, 1087-1092.
- [27] Apaydin, M.S.; Brutlag, D.L.; Guestrin, C.; Hsu, D.; Latombe, J.C. Proc RECOMB 2002, 12-21.
- [28] Ramachandran, G.N.; Sasisekharan, V. Adv Prot Chem 1968, 23, 283-438.
- [29] LaValle, S.M.; Finn, P.W.; Kavradi, L.E.; Latombe, J.C. J Comp Chem 2000, 21, 731-747.
- [30] Scott, R.A.; Scheraga, H.A. J Chem Phys 1966, 44, 3054-3069.
- [31] Craig, J.J. Introduction to Robotics: Mechanics and Control; Addison-Wesley, 1989; Chapter 3.
- [32] Zhang, M.; Kavradi, L.E. J Chem Info and Comp Sci 2002, 42(1), 64-70.
- [33] Edelsbrunner, H. Proc WAFR 1998, 265-277.

- [34] Bondi, A. J Phys Chem 1964, 68, 441-451.
- [35] Nielsen, J.; Roth, B. Proc NATO Adv Study Inst Comp Meth Mech 1997, 1, 233-252.
- [36] Lee, H.Y.; Liang, C.G. Int J Mech and Mach Theory 1988, 23(3), 209-217.
- [37] Raghavan, M.; Roth, B. Proc Int Symp Rob Res 1989, 314-320.
- [38] Manocha, D.; Canny, J. IEEE Trans Rob & Autom 1994, 10(5), 648-657.
- [39] Renaud, M. Proc Int Conf Mech Design & Prod 2000, 15-25.
- [40] Crippen, G.M. J Comp Chem 1992, 13, 351-361.
- [41] Celaya, E.; Torras, C. in: Computational Kinematics; Kluwer Academic Publishers, 1993; pp. 85-94.
- [42] Ricard, R.; Gosselin, C.M. J Mech Design 1998, 120(2), 269-278.
- [43] Merlet, J.P. J Intell and Rob Sys 1995, 13, 143-160.
- [44] Pauling, L. The Nature of the Chemical Bond; Cornell University Press, 1960.
- [45] Angeles, J.; Fundamentals of Robotic Mechanical Systems; Springer-Verlag, 2003.
- [46] Kwakernaak, H.; Sebek, M.; IEEE Trans Autom Control 1994, 39(2), 315-328.
- [47] Lin, M.; Manocha, D.; Cohen, J.; Gottschalk, S. in: Algorithms for Robotic Motion and Manipulation; A.K. Peters, 1997; pp. 129-141.
- [48] Jiménez, P.; Thomas, F.; Torras, C. in: Robot Motion Planning and Control; Springer-Verlag, 1998; pp. 305-343.
- [49] van Geem, C.; Siméon, T. LAAS Report N°01073, 2001.
- [50] Burdick, J.W. Proc IEEE Int Conf Rob & Autom 1989, 264-270.
- [51] Cortés, J.; PhD Thesis; LAAS Report, 2003.

- [52] Bruccoleri, R.E.; Karplus, M. *Biopolymers* 1985, 18, 2767-2773.
- [53] Skov, L.K.; Mirza, O.; Henriksen, A.; de Montalk, G.P.; Remaud-Simeon, M.; Sarçabal, P.; Willemot, R.M.; Monsan, P.; Gajhede, M. *J Biol Chem* 2001, 276, 25273-25278.
- [54] Skov, L.K.; Mirza, O.; Sprogø, D.; Dar, I.; Remaud-Simeon, M.; Albenne, C.; Monsan, P.; Gajhede, M. *J Biol Chem* 2002, 277, 47741-47747.
- [55] Derreumaux, P.; Schlick, T.; *Biophys J* 1998, 74, 72-81.
- [56] Siméon, T.; Laumond, J.P.; Lamiroux, F. *Proc IEEE Int Conf Rob & Autom* 2001, 25-30.
- [57] Bruccoleri R.E.; Haber, E.; Novotny, J. *Nature* 1988, 335, 564-568.
- [58] Amato, N.M.; Dill, K.A.; Song, G. *J Comp Biol* 2002, 9(2), 149-168.
- [59] Apaydin, M.S.; Singh, A.P.; Brutlag, D.L.; Latombe, J.C. *Proc IEEE Int Conf Rob & Autom* 2001, 932-939.
- [60] Bayazit, O.B.; Song, G.; Amato, N.M. *Proc IEEE Int Conf Rob & Autom* 2001, 954-959.

Figure Captions

Figure 1: Molecular chain model. Frames associated with atoms (a) and defining the articulated mechanism (b).

Figure 2: Steps of the RLG algorithm performed on a $6R$ planar linkage.

Figure 3: Maximum extension of polypeptide backbone with Pauling-Corey geometry.

Figure 4: Incremental exploration of \mathcal{C}_{feas} using an RRT-like technique.

Figure 5: Structure of amylosucrase from *Neisseria polysaccharea*.

Figure 6: Exploration (a) with and (b) without the side-chain of Asp²³¹.

Figure 7: Simulated conformational gating of loop 7 in amylosucrase.

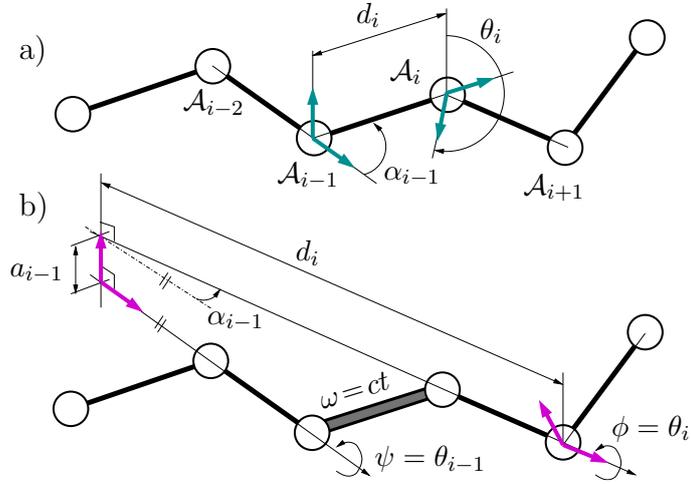


Figure 1: Molecular chain model. Frames associated with atoms (a) and defining the articulated mechanism (b).

Algorithm 1: RANDOMLOOPCONF

```

input   : the loop, the rest of the protein
output  : the conformation  $q$ 
begin
   $q_b \leftarrow \text{RANDOMBACKBONECONF}(\text{loop.bkb});$ 
  if not CLASHCHECK( $q_b$ , loop.bkb, protein) then
    if  $q_s \leftarrow \text{GENERATESIDECHAINS}(q_b, \text{loop}, \text{protein})$  then
       $q \leftarrow \text{COMPOUNDCONF}(\text{loop}, q_b, q_s);$ 
    else return Failure;
  else return Failure;
end

```

Algorithm 2: RANDOMBACKBONECONF

```

input   : the backbone
output  : the conformation  $q_b$ 
begin
   $q^a \leftarrow \text{SAMPLE}_q^a(\text{backbone});$ 
  if  $q^p \leftarrow \text{COMPUTE}_q^p(\text{backbone}, q^a)$  then
     $q_b \leftarrow \text{COMPOUNDCONF}(\text{backbone}, q^a, q^p);$ 
  else return Failure;
end

```

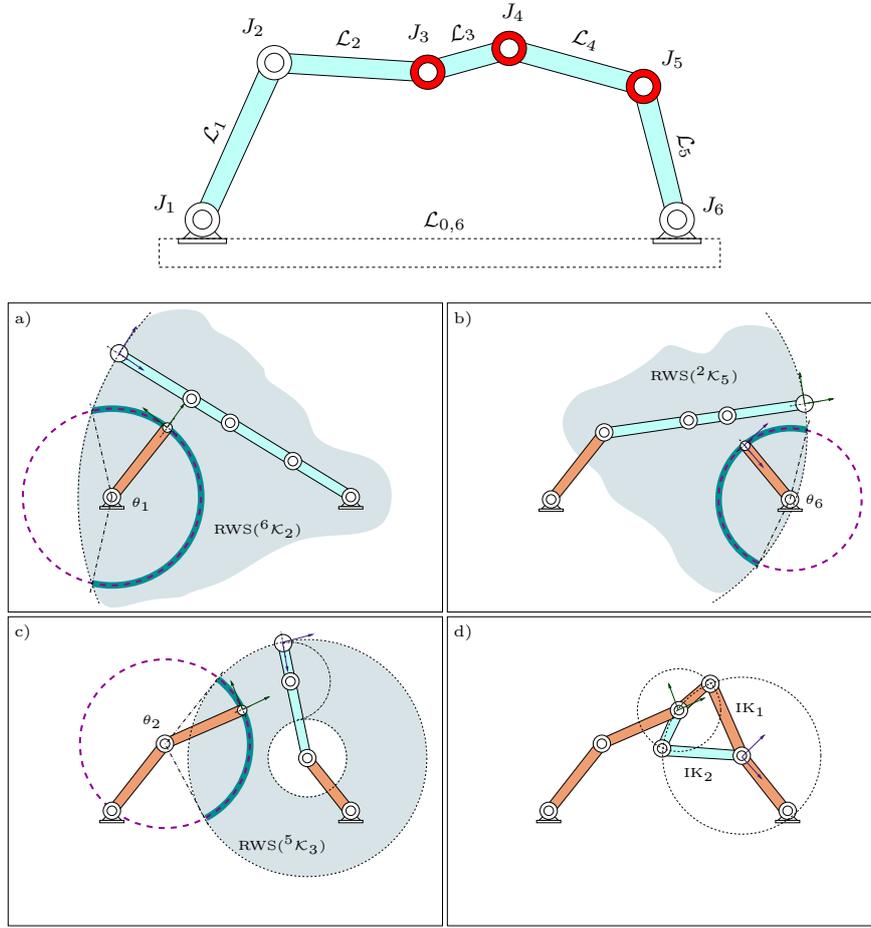


Figure 2: Steps of the RLG algorithm performed on a 6R planar linkage.

Algorithm 3: $\text{SAMPLE_}q^a$

```

input   : the backbone
output  : the active variables  $q^a$ 
begin
1 |  $(J_b, J_e) \leftarrow \text{INITSAMPLER}(\textit{backbone});$ 
   | while not  $\text{ENDACTIVECHAIN}(\textit{backbone}, J_b)$  do
   | |  $I_c \leftarrow \text{COMPUTECLOSURERANGE}(\textit{backbone}, J_b, J_e);$ 
   | | if  $I_c = \emptyset$  then goto line 1;
   | |  $\text{SETJOINTVALUE}(J_b, \text{RANDOM}(I_c));$ 
   | |  $J_b \leftarrow \text{NEXTJOINT}(\textit{backbone}, J_b);$ 
   | | if not  $\text{ENDACTIVECHAIN}(\textit{backbone}, J_e)$  then
   | | |  $\text{SWITCH}(J_b, J_e);$ 
   | end
end

```

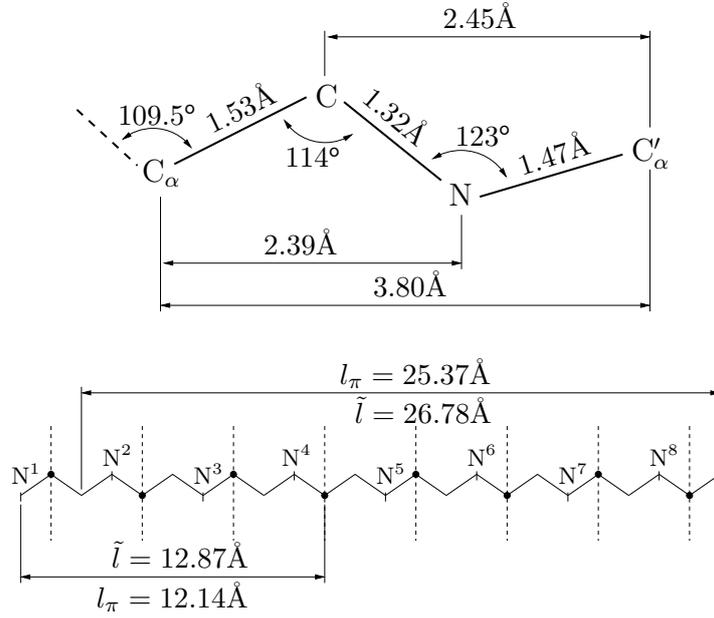


Figure 3: Maximum extension of polypeptide backbone with Pauling-Corey geometry.

Algorithm 4: ExploreByRRT

```

input   : the loop, the rest of the protein,  $q_{init}$ 
output : the tree  $\mathcal{T}$ 
begin
   $G \leftarrow \text{INITTREE}(q_{init});$ 
   $n_{fail} \leftarrow 0;$ 
  while not STOPCONDITION( $\mathcal{T}$ ) do
     $q_{rand} \leftarrow \text{GUIDEDRANDOMCONF}(\text{loop});$ 
     $q_{near} \leftarrow \text{NEARESTNEIGHBOR}(q_{rand}, \mathcal{T});$ 
     $q_{feas} \leftarrow q_{near};$ 
     $state \leftarrow OK;$ 
    while  $state = OK$  do
       $q_{step} \leftarrow \text{MAKESTEP}(q_{feas}, q_{rand});$ 
      if FEASIBLECONF( $q_{step}$ ) then  $q_{feas} \leftarrow q_{step};$ 
      else  $state \leftarrow FAIL;$ 
    if not TOOSIMILARCONF( $q_{near}, q_{feas}$ ) then
       $q_{new} \leftarrow \text{INTERMEDIATECONF}(q_{near}, q_{feas});$ 
      GROWTREE( $q_{new}, q_{near}, \mathcal{T}$ );
       $n_{fail} \leftarrow 0;$ 
    else  $n_{fail} \leftarrow n_{fail} + 1;$ 
  end

```

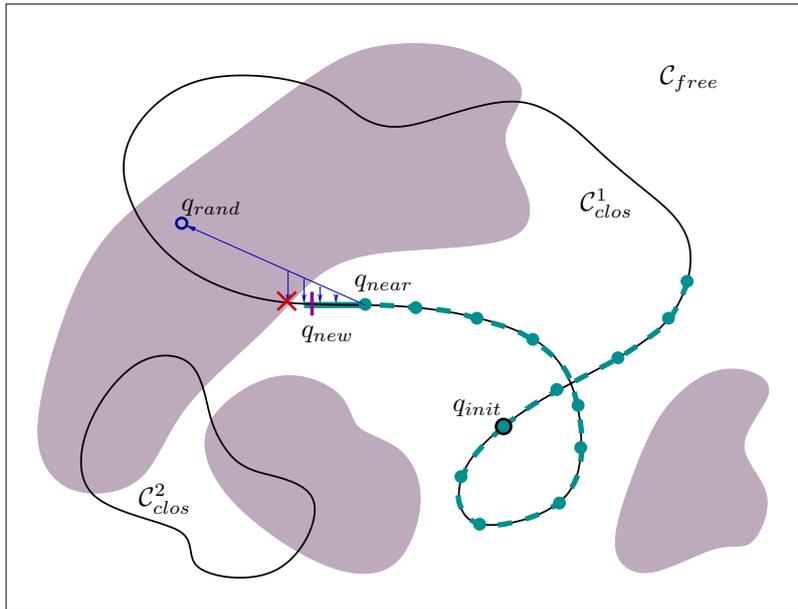


Figure 4: Incremental exploration of C_{feas} using an RRT-like technique.

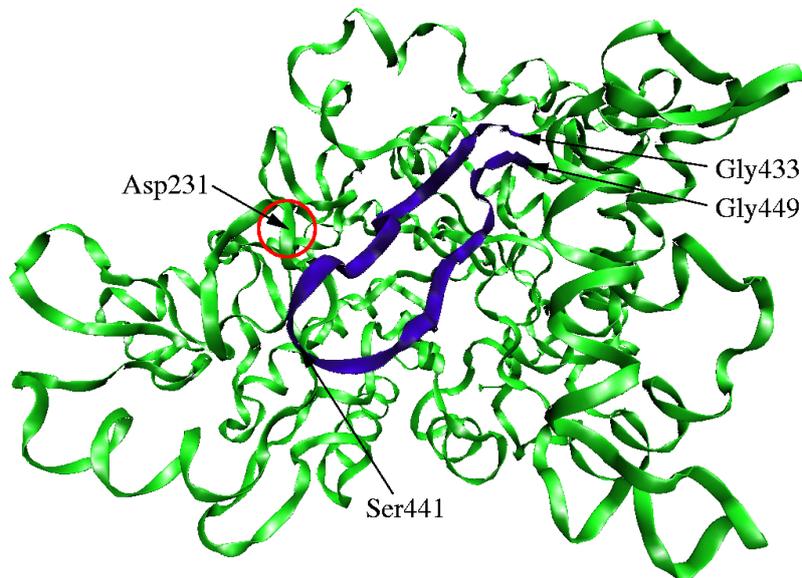


Figure 5: Structure of amylosucrase from *Neisseria polysaccharea*.

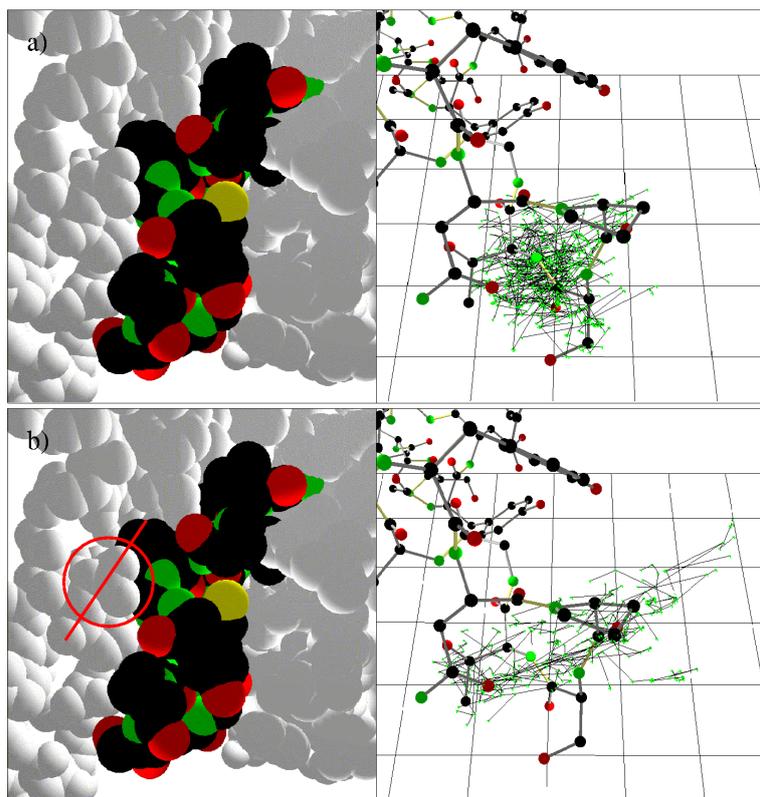


Figure 6: Exploration (a) with and (b) without the side-chain of Asp²³¹.

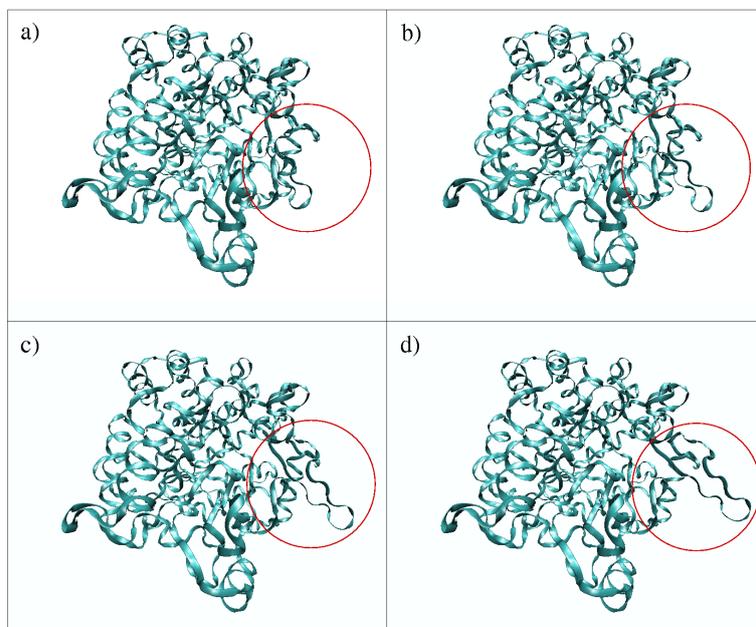


Figure 7: Simulated conformational gating of loop 7 in amylosucrase.