Efficient Algorithms to Explore Conformation Spaces of Flexible Protein Loops

Peggy Yao*, Ankur Dhanik*, Nathan Marz*, Ryan Propper*, Charles Kou*, Guanfeng Liu*,
Henry van den Bedem[†], Jean-Claude Latombe*, Inbal Halperin Landsberg[‡], Russ B. Altman[‡]
*Computer Science Department, Stanford University, Stanford, CA 94305, USA
[†]Joint Center for Structural Genomics, SLAC, Menlo Park, CA 94025, USA
[‡]Genetics Department, Stanford University, Stanford, CA 94305, USA
Email:peggyyao@stanford.edu

***Abstract*—Several applications in biology – e.g., incorporation of protein flexibility in ligand docking algorithms, interpretation of fuzzy X-ray crystallographic data, and homology modeling – require computing the internal parameters of a flexible fragment (usually, a loop) of a protein in order to connect its termini to the rest of the protein without causing any steric clash inside the loop and with the rest of the protein. One must often sample many such conformations in order to explore and adequately represent the conformational range of the studied loop. While sampling must be fast, it is made difficult by the fact that two conflicting constraints – kinematic closure and clash avoidance – must be satisfied concurrently. This paper describes two efficient and complementary sampling algorithms to explore the space of closed clash-free conformations of a flexible protein loop. The "seed sampling" algorithm samples broadly from this space, while the "deformation sampling" algorithm uses seed conformations as starting points to explore the conformation space around them at a finer grain. Computational results are presented for various loops ranging from 5 to 25 residues. More specific results also show that the combination of the sampling algorithms with a functional site prediction software (FEATURE) makes it possible to compute and recognize calcium-binding loop conformations. The sampling algorithms are implemented in a toolkit, called LoopTK, which is available at** `https://simtk.org/home/looptk`**.**

I. Introduction

Several applications in biology require *exploring* the conformation space of a flexible fragment (usually, a loop) of a protein. For example, upon binding with a small ligand, a fragment may undergo deformations to rearrange non-local contacts [22]. Incorporating such flexibility in docking algorithms is a major challenge [25]. In X-ray crystallography experiments, electron-density maps often contain noisy regions caused by disorder in the crystalline sample, resulting in an initial model with missing fragments between resolved termini [27]. Similarly, in homology modelling [23], only parts of a protein structure can be reliably inferred from known structures with similar sequences. These applications share a common sub-problem: to compute closed, clash-free conformations of an inner fragment of a protein chain. These conformations lie in a complex subset of the fragment's conformation space.

This problem requires satisfying two constraints concurrently: closing a kinematic loop and avoiding steric clashes. Each constraint considered separately is relatively easy to satisfy, but the combination is hard because the two constraints are conflicting. The closed conformations of a loop with n degrees of freedom (DOFs) – e.g., n dihedral angles ϕ and ψ – form a subspace of dimensionality at least $n-6$ contained in the n-dimensional conformation space of the loop. Due to protein compactness, the conformations that are both closed and clash-free typically form a subset of this subspace that has a very small relative volume, especially for long loops. Hence, an arbitrary closed conformation of the loop has small probability to be clash-free. Conversely, an arbitrary collision-free conformation of the loop has null probability to be closed. As a result, existing sampling techniques often have high rejection ratios.

In this paper, we present two new techniques, *seed* and *deformation* sampling, to solve this problem. Each deformation sampling operation starts from a given closed clash-free conformation (a "seed") and deforms this conformation without breaking closure or introducing clashes by modifying the loop's DOFs in a coordinated way. In contrast, seed sampling generates new conformations from scratch, by prioritizing the treatment of the two constraints, so that the most limiting one is enforced first. In both techniques, prevention and detection of steric clashes is done using the grid-indexing method described in [16]. Seed and deformation sampling complement each other very well. Seed sampling produces conformations that are broadly distributed over the loop's conformation space and provides conformations (seeds) later used by deformation sampling to explore more finely certain regions of this space. These algorithms are implemented into a toolkit, **LoopTK**, available at `https://simtk.org/home/looptk`. They have been tested on various loops ranging from 5 to 25 residues.

Section II compares our work to motivation and previous work. Section III outlines the loop kinematic model used in this paper. Sections IV and V describe the seed and deformation sampling algorithms, respectively. Section VI briefly presents the grid technique used both to detect steric clashes and to identify pairs of close atoms. Section VII discusses various results obtained with the implemented software. In particular, Section VII-D shows that the combination of our algorithms with FEATURE (a functional site prediction software) [30], makes it possible to compute calcium-binding loop conformations.

II. Motivation and Previous Work

The problem considered in this paper is a version of the "loop closure" problem studied in [4], [9], [11], [17], [20], [29]. Several works have specifically focused on kinematic closure. Analytical Inverse Kinematics (IK) methods are described in [9], [29] to close a fragment of 3 residues. For longer fragments, iterative techniques have been proposed, like the popular CCD (Cyclic Coordinate Descent) [4] and the "null space" technique [24], [27]. We re-use several of these techniques in our work. In particular, our seed sampling algorithm applies the analytical IK method described in [9] in a new way to close loops with more than 3 residues. Our deformation sampling algorithm uses the null space technique to deform loops without breaking closure.

Several sampling procedures have been proposed to generate closed clash-free conformations of loops, in particular [8], [11], [17].

- RAPPER [11] iteratively builds up a loop conformation from its N terminus toward its C terminus. It selects the values of the dihedral angles ϕ and ψ in each successive residue at random from a predefined table of values. It also checks that the Cα atom in each residue is sufficiently close to the loop's C anchor on the protein. However, when a complete conformation has been generated, there remains a potentially large gap between the loop's last residue and its anchor on the protein. So, RAPPER runs an iterative minimization procedure to reduce this gap. Overall, RAPPER has the same purpose as our seed sampling algorithm. But our algorithm differs in several ways: in particular, the angles ϕ and ψ are not selected from discrete tables and a sufficient number of dihedral angles are retained (in the middle portion of the loop) to make it possible to apply an exact IK method.
- Our seed sampling algorithm is more similar to the hierarchical approach proposed in [17]. Both methods decompose a loop into three fragments, independently sample clash-free conformations of the two fragments respectively rooted at the N and C termini, and finally close the loop with the middle fragment. However, like RAPPER, the method in [17] selects the values of the ϕ and ψ angles from predefined discrete sets, which may make it more difficult or impossible to build clash-free conformations in very constrained areas. In addition, the IK and steric clash detection methods it uses are very different from ours. Both the method in [17] and RAPPER were tested on relatively short loops having 2 to 12 residues in length.
- RLG [8] successively samples closed conformations that it later tests for steric clashes. To sample closed conformations it divides the loop backbone into an "active" and a "passive" fragment. The latter has exactly 3 residues. The active fragment is progressively sampled at random using a geometric algorithm that increases the likelihood that a closed conformation will eventually be obtained. The 6 dihedral angles of the passive fragment are used to close the loop using an IK procedure. The generated closed conformations are then tested for steric clashes. Except for short loops this type of procedure has a high rejection ratio, especially when clash-free conformations span a relatively small subset of the closed conformation space, which is the case for most long loops. This observation motivated the prioritized constraint-satisfaction approach embedded in our seed sampling procedure.

All three methods discussed above try to solve the same problem as our seed sampling algorithm. However, seed sampling inherently suffers from an obvious dilemma. On the one hand, it should generate conformations broadly distributed across the closed clash-free conformation space of a loop. On the other hand, it should have the ability to sample biologically interesting conformations, for instance near-native conformations. Except for short loops, these two goals are conflicting. This is why our algorithms also include a deformation sampling method to generate denser sets of conformations around selected seed conformations.

Some sampling procedures try to sample conformations using libraries of fragments obtained from previously solved structures [10], [20], [26], [28]. For example, a divide and conquer approach is described in [26] that generates a database of fragments of different residue lengths and types, by using a Ramachandran plot distribution. These fragments are then concatenated to build conformations of a longer loop. However, steric clashes are not taken into account during this process.

Other works sample conformations by minimizing an energy function [2], [11], [12], [17], [24] or running a molecular dynamics simulation [3] with the goal to identify loop fragments close to native structure. However, in the case of a truly deformable loop, it is often more useful to explore the entire closed clash-free conformation space. For example, a fuzzy electron density map may be better explained by an ensemble of conformations than by a single one [13], [21]. Our goal in this paper is to present such exploration tools. Nevertheless, our deformation sampling technique also allows energy minimization, when this is desirable. We show in Section VII-D that our sampling procedures can generate biologically interesting conformations.

In our algorithms, steric clash detection is done using the efficient grid method previously described in [16]. A similar detection method is also used in RAPPER [11].

III. Loop Model

A *loop* L is defined here as a sequence of $p > 3$ consecutive residues in a protein P, such that none of the two termini of L is also a terminus of P. We number the residues of L from 1 to p, starting from the N terminus. We model the backbone of L as a serial linkage whose DOFs are the $n = 2p$ dihedral angles ϕ_i and ψ_i around the bonds N–Cα and Cα–C, in residues $i = 1, ..., p$. The rest of the protein, denoted by $P \backslash L$, is assumed rigid. We let L_B denote the backbone of L. It includes the Cβ and O atoms respectively bonded to the Cα and C atoms in the backbone.

We attach a Cartesian coordinate frame Ω_1 to the N terminus of L and another frame Ω_2 to its C terminus. When L_B is connected to its anchors in the rest of the protein, i.e., when it adopts a *closed* conformation, the pose (position and orientation) of Ω_2 relative to Ω_1 is fixed to a predefined value that we denote by Π_g.

If we arbitrarily pick the values of ϕ_i and ψ_i, $i = 1$ to p, then in general we get an *open* conformation of L_B, where the pose of Ω_2 relative to Ω_1 differs from Π_g. The set $\mathbf{Q}$ of all open and closed conformations of L_B is a space of dimensionality $n = 2p$. The subset $\mathbf{Q}_{\text{closed}}$ of closed conformations is a subspace of $\mathbf{Q}$ of dimensionality at least $n - 6$. Let $\Pi(q)$ denote the pose of Ω_2 relative to Ω_1 when the conformation of L_B is $q \in \mathbf{Q}$. The function Π and its inverse Π^{-1} are the "forward" and "inverse" kinematics map of L_B, respectively.

A conformation of L_B is *clash-free* if and only if no two atoms, one in L_B, the other in L_B or $P \backslash L$, are such that their centers are closer than ε times the sum of their van der Waals radii, where ε is a constant in $(0, 1)$. In our software, ε is an adjustable parameter, usually set to 0.75, which approximately corresponds to the distance where the van der Waals potential associated with two atoms begins increasing steeply. We denote the set of closed clash-free conformations of L_B by $\mathbf{Q}^{\text{free}}_{\text{closed}}$. In general, it has the same dimensionality as $\mathbf{Q}_{\text{closed}}$, but its volume is usually a small fraction of that of $\mathbf{Q}_{\text{closed}}$.

IV. Seed Sampling

A. Overview

The goal of seed sampling is to generate conformations of L_B broadly distributed over $\mathbf{Q}^{\text{free}}_{\text{closed}}$. The challenge comes from the interaction between the kinematic closure and clash avoidance constraints. Computational tests (see Section VII) show that the approach (hereafter called the *naive* approach) that first samples conformations from $\mathbf{Q}_{\text{closed}}$ and next rejects those with steric clashes is often too time consuming, except for short loops, due to its huge rejection ratio. The reverse approach – sampling the angles ϕ_i and ψ_i of L_B to avoid clashes – will inevitably end up with open conformations, since $\mathbf{Q}_{\text{closed}}$ has lower dimensionality than $\mathbf{Q}$.

These insights led us to develop a prioritized constraint-satisfaction approach, hereafter called the *prioritized* approach. We partition L_B into three segments, the front-end F, the mid-portion M, and the back-end B. F starts at the N terminus of L_B and B ends at its C terminus. M is the segment between them. Due to the immediate proximity of atoms in $P \backslash L$, the conformations of F and B are more limited by the clash avoidance constraint than by the closure constraint; so, we sample the dihedral angles in F and B to avoid clashes, ignoring the closure constraint. Then, for any pair of conformations of F and B, the possible conformations of M are mainly limited by the closure constraint; so, we use the naive approach to sample conformations of M, by running an IK procedure to close the gap between F and B and testing the clash avoidance constraint afterward. In this way, our prioritized approach reduces the application of the naive approach to a short fragment of the loop. The length of M must be large enough for the IK procedure to succeed with high probability, but not too large since clash avoidance is only tested afterward. In our software, the number of residues in M is usually set to half of that of L_B or to 4, whichever of these two numbers is larger. The number of residues of F and B are then selected equal (± 1). Tests show that these choices are close to optimal on average for a wide range of loops. For unusually long loops, it may be suitable to set an upper bound on the length of M.

The dihedral angles ϕ and ψ in the three fragments F, M, and B are selected to generate conformations of L_B broadly distributed over $\mathbf{Q}^{\text{free}}_{\text{closed}}$.

B. Sampling front/back-end conformations

Consider the front-end F. The angles ϕ and ψ closest to the fixed terminus of F are the most constrained by possible clashes with the rest of the protein $P \backslash L$. So, the angles are sampled in the order in which they appear in F, that is ϕ_1, ψ_1, ϕ_2, etc. In this order, each angle ϕ_i (resp., ψ_i) determines the positions of the next two atoms $\text{C}_{\beta i}$ and C_i (resp., the next three atoms O_i, N_{i+1} and $\text{C}\alpha_{i+1}$). The angle is sampled so that these atoms do not clash with any atom in $P \backslash L$ or any preceding atom in F. Its value is picked at random, either uniformly or according to a user-input probabilistic distribution (e.g., one based on Ramachandran tables). If no value of the angle prevents the two or three atoms it governs from clashing with other atoms, the algorithm backtracks and re-samples a previously sampled angle. Clash-free conformations of the back-end B are sampled in the same way, by starting from its fixed C terminus and proceeding backward.

C. Sampling mid-portion conformations

Given two non-clashing conformations of F and B such that the gap between them does not exceed the maximal length that M can achieve, a conformation of M is sampled as follows.

The values of the ϕ and ψ angles in M are picked at random, uniformly or according to a given distribution. This leads to a conformation q of M that is connected to F at one end and open at the other end. To close the gap between M and B, we use the IK method described in [9]. This method solves the IK problem analytically, for any sequence of residues in which exactly three pairs of (ϕ, ψ) dihedral angles are allowed to vary. These pairs need not be consecutive.

Let us denote the IK method by ANALYTICAL-IK(q, i, j, k), where argument q is the initial open conformation of M and arguments i, j, and k are the integers identifying the three residues that contain the pairs of dihedral angles that are allowed to vary. Our experiments show that, on average, the IK method is the most likely to succeed in closing the gap when one pair is the last one in M and the other two are distributed in M. Let r and s denote the integers identifying the first and last residue of M in L_B. As the IK method is extremely fast, ANALYTICAL-IK(q, i, j, s) is called for all $i = r, ..., s - 2$ and $j = i + 1, ..., s - 1$, in a random order, until a closed conformation of M has been generated. If this conformation tests clash-free, then the seed sampling procedure constructs a closed clash-free conformation of L_B by concatenating the conformations of F, M, and B.

If the above operations fail to generate a closed clash-free conformation of M, then they are repeated (with new initial values for the ϕ and ψ angles in M) until a predefined maximal number of iterations have been performed.

We have also experimented with iterative IK techniques, like CCD, to close the gap between M and B. In our implementation they were slower than the above algorithm based on analytical IK.

D. Placing side-chains

For each conformation of L_B sampled from $\mathbf{Q}^{\text{free}}_{\text{closed}}$, we use SCWRL3 [5] to place the side-chains. We may only compute the placements of the side-chains in L_B given the placements of the side-chains in $P \backslash L$. Alternatively, we may (re-)compute the placements of all the side-chains in the protein. In each case, SCWRL3 minimizes an energy function that contains volume-exclusion terms. But it does not fully guarantee that the conformations of the side-chains will be clash-free. If needed, we can use deformation sampling to slightly deform the conformation of L_B in order to eliminate the steric clashes (see Section VII-C).

V. Deformation Sampling

A. Overview

The deformation sampling procedure is given a "seed" conformation q in $\mathbf{Q}^{\text{free}}_{\text{closed}}$. It first selects a vector in the tangent space $T\mathbf{Q}_{\text{closed}}(q)$ of $\mathbf{Q}_{\text{closed}}$ at q. By definition, any vector in this space is a velocity vector $[\dot{\phi}_1, ..., \dot{\psi}_n]^T$ that maps to the null velocity of Ω_2 (relative to Ω_1); hence, it defines a direction of motion that does not instantaneously break loop closure. A new conformation of L_B is then computed as $q' = q + \delta q$ where δq is a short vector in $T\mathbf{Q}_{\text{closed}}(q)$. Since the tangent space is only a local linear approximation of $\mathbf{Q}_{\text{closed}}$ at q, the closure constraint is in fact slightly broken at q'. So, ANALYTICAL-IK(q', $p-2$, $p-1$, p) is called to bring back the frame Ω_2 to its goal pose Π_g. Since q' is already almost closed, the six DOFs used by ANALYTICAL-IK are the angles ϕ_{p-2}, ... ψ_p corresponding to the last three residues of L_B (recall that $n = 2p$). If ANALYTICAL-IK generates several solutions for these angles, the closest values from those in $q + \delta q$ are selected. Finally, the atoms in L_B are tested for clashes among themselves and with the rest of the protein. If a clash is detected, the procedure exits with failure.

The deformation sampling procedure may be run several times with the same seed conformation q to explore the subset of $\mathbf{Q}^{\text{free}}_{\text{closed}}$ around q. Alternatively, each run may use the conformation generated at the previous run as the new seed to generate a "pathway" in the set $\mathbf{Q}^{\text{free}}_{\text{closed}}$. More generally, one may also build a tree of pathways rooted at a seed conformation or a forest of trees rooted at multiple seeds, e.g. to optimize an objective function.

B. Computation of a basis of the tangent space

To define a direction in $T\mathbf{Q}_{\text{closed}}(q)$, we must first compute a basis for this space. This can be done as follows [27]. Let $J(q)$ be the $6 \times n$ Jacobian matrix that maps the velocity $\dot{q} = [\dot{\phi}_1, ..., \dot{\psi}_p]^T$ of the dihedral angles in L_B at q to the velocity $[\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma}]^T$ of Ω_2, i.e.: $[\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma}]^T = J(q)\dot{q}$. $J(q)$ can be computed analytically using techniques presented in [7]. For simplicity, assume that J has full rank (i.e., 6). A basis of $T\mathbf{Q}_{\text{closed}}(q)$ is built by first computing the Singular Value Decomposition ($U\Sigma V^T$) of $J(q)$ where U is a 6×6 unitary matrix, Σ is a $6 \times n$ matrix with non-negative numbers on the diagonal and zeros off the diagonal, and V is an $n \times n$ unitary matrix [15]. Since the rows $6, ..., n$ of V do not affect the product $J(q)\dot{q}$, their transposes form an orthogonal basis $N(q)$ of $T\mathbf{Q}_{\text{closed}}(q)$.

C. Selection of a direction in the tangent space

The deformation sampling procedure may select a direction in $T\mathbf{Q}_{\text{closed}}(q)$ at random. However, in most cases, it is preferable to minimize an objective function $E(q)$. Let $y = -\nabla E(q)$ be the negated gradient of E at q and $y_N = NN^T y$ the projection of y into $T\mathbf{Q}_{\text{closed}}(q)$. The deformation sampling procedure selects the increment δq along y_N. In this way, all the DOFs left available in L_B by the closure constraints are used to move the conformation in the direction that most reduces E.

$E(q)$ may be a function of the distances between the closest pairs of atoms at conformation q (where each pair consists of one atom in L_B and one atom in either $L \backslash B$ or L_B). These pairs can be efficiently computed by the same grid method that is used to detect steric clashes (Section VI). Minimizing E then leads deformation sampling to increase the distances between these pairs of atoms, if this goal does not conflict with the closure constraint. In this way, deformation sampling picks increments δq that have small risk of causing steric clashes.

Another interesting objective function leads to moving a designated atom A in L_B toward a desired position x_d. This objective function can be defined as:

$$E(q) = \|x_A(q) - x_d\|^2. \tag{1}$$

where $x_A(q)$ is the position of A when L_B's conformation is q. This function can be used to iteratively move an atom as far as possible along selected directions to explore the boundary of $\mathbf{Q}^{\text{free}}_{\text{closed}}$. E can also be an energy function or any weighted combination of functions, each designed to achieve a distinct purpose.

D. Placing side-chains

For each new conformation of L_B, side-chains can be placed using SCWRL3, as described in Section IV. Another possibility is to provide an initial seed conformation that already contains the loop's side-chains to the deformation sampling procedure. These side-chains are then considered rigid and the procedure deforms L_B so that the produced conformation remains clash-free.

VI. Steric Clash Detection

Steric clash detection is done using the grid method [16]. This method takes advantage of the fact that, to avoid clashes, atoms must spread out, so that any square box of a fixed volume contains an upper-bounded number of atom centers, independent of the total number of atoms in the protein.

The method tessellates the three-dimensional space of the protein into an array of equally sized cubes. The edge length of a cube is chosen approximately equal to the largest diameter of the atoms. For a given conformation of the protein, each atom is indexed in the cube that contains its center. Whenever the position of an atom is modified, the grid structure is updated accordingly in constant time. The grid is implemented as a memory-efficient hash table. Only the grid cubes that contain atom centers are represented, each with the corresponding list of atoms.

The clash detection algorithm iterates through all atoms that need to be checked (e.g., the atoms in L_B), asking for each atom if it is in collision. The atom only needs to be checked with the atoms indexed in its own grid cube and the 26 cubes surrounding it. Since the cubes of the grid are small, this amounts to only checking a few pairs of atoms (usually less than 6). Consequently, clash detection for a single atom runs in $\mathcal{O}(1)$ time, and the clash test for all $\mathcal{O}(n)$ atoms in L_B or L runs in $\mathcal{O}(n)$ time, independent of the total number of atoms in the protein. The same algorithm can be used to find the k closest atoms to a given atom (for a small value of k), simply by considering another layer of grid cubes. This ability allows us to efficiently compute objective functions E, like the one in Eq. (1) that contains terms aimed at preventing deformation sampling from producing conformations with steric clashes (Section V-C).

VII. Results

A. Seed sampling

Table I lists 20 loops, whose sizes range from 5 to 25 residues, which we used to perform computational tests. Each row lists the PDB id of the protein, the number of residues in the protein, the number identifying the first residue in the loop, the number of residues in the loop, and the average time to sample one closed clash-free conformation of the loop using two distinct procedures (our seed sampling method and the "naive" method outlined in Section IV-A). In some loops the two termini are close, while in others they are quite distant. Some loops protrude from the proteins and have much empty space in which they can deform without clash (e.g., 3SEB), while others are very constrained by the other protein residues (e.g., 1TIB). The loop in 1MPP is constrained in the middle by side-chains protruding from the rest of the protein (see Figure 2(b)). In the results presented below, all ϕ and ψ angles were picked uniformly at random (i.e., no biased distributions, like the Ramachandran's ones, were used).

Each picture in Figure 1 displays a subset of backbone conformations generated by seed sampling for the loops in 1TIB, 3SEB, 8DFR, and 1THW. The loop in 1TIB, which resides at the middle of the protein, has very small empty space to move in. The PDB conformation of the loop in 1THW (shown green in the picture) bends to the right, but our method also found clash-free conformations that are very different. Each picture in Figure 2 shows the distributions of the middle Cα atom in 100 sampled conformations of the loops in proteins 1K8U, 1MPP, 1COA, and 1G5A along with a few backbone conformations. The loops in 1K8U and 1COA have relatively large empty space to move in, whereas the loops in 1MPP and 1G5A are restricted by the surrounding protein residues. These figures illustrate the ability of our seed sampling procedure to generate conformations broadly distributed across the closed clash-free conformation space of a loop.

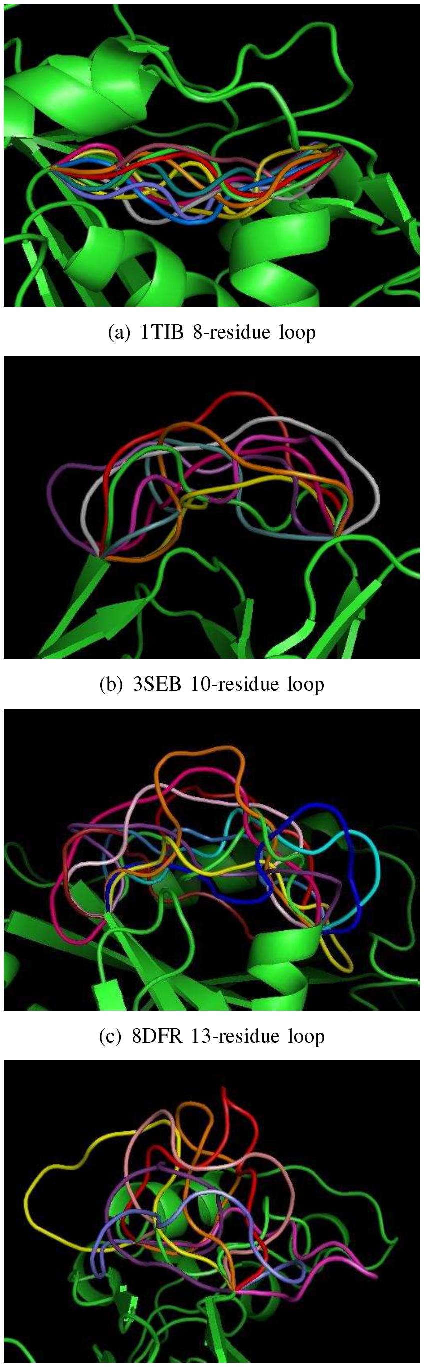
Protein		Loop		Sampling	
Id	Size	Start	Size	Seed	Naive
1XNB	185	SER 31	5	0.22	0.21
1TYS	264	THR 103	5	0.06	0.06
1GPR	158	SER 74	6	0.38	0.38
1K8U	89	GLU 23	7	0.21	0.20
2DRI	271	GLN 130	7	0.42	0.46
1TIB	269	GLY 172	8	2.49	13.03
1PRN	289	ASN 215	8	0.33	0.66
1MPP	325	ILE 214	9	0.53	99.85
4ENL	436	LEU 136	9	1.46	19.35
135L	129	ASN 65	9	0.77	1.54
3SEB	238	HIS 121	10	0.50	3.80
1NLS	237	ASN 216	11	1.30	5.51
1ONC	103	MET 23	11	2.26	5.66
1COA	64	VAL 53	12	19.02	67.49
1TFE	142	GLU 158	12	0.48	8.14
8DFR	186	SER 59	13	2.02	39.36
1THW	207	CYS 177	14	1.48	9.84
1BYI	224	GLU 115	16	2.52	>800
1G5A	628	GLY 433	17	3.28	>800
1HML	123	GLY 51	25	17.74	>800

TABLE I
Testset of 20 loops (see main text for comments).

The average running time (in seconds) of our seed sampling procedure to compute one closed clash-free conformation of each loop is shown in column 5 of Table I. Each average was obtained by running the procedure until it generated 100 conformations of the given loop and dividing the total running time by 100.[1] The last column of Table I gives the average running time of the "naive" procedure that first samples closed conformations of the loop backbone and next rejects those which are not clash-free. In both procedures, the factor ε used to define steric clashes (see Section III) was set to 0.75. Our seed sampling procedure does not break a loop into 3 segments if it has fewer than 8 residues. So, the running times of both procedures for the first 5 proteins are essentially the same. For all other proteins, our procedure is faster, sometimes by a large factor (188 times faster for the highly constrained loop in 1MPP), than the naive procedure. For the last three proteins, this latter procedure failed to sample 100 conformations after running for more than 80,000 seconds.

Not surprisingly, the running times vary significantly across loops. Short loops with much empty space around them take a few 1/10 seconds to sample, while long loops with little empty space can take a few seconds to sample. The loops in 1COA and 1HML take significantly more time to sample than the others. In the case of 1COA, it is difficult to connect the loop's front-end and back-end (3 residues each) with its mid-portion (6 residues). As Figure 6 shows, the termini of the loop are far apart and the protein constrains the loop all along. Due to the local shape of the protein at the two termini of the loop, many sampled front-ends and back-ends tend to point in opposite directions, which then makes it often impossible to close the mid-portion without clashes. In this case, we got a better average running time (4 seconds, instead of 19) by setting the length of the mid-portion to 8 (instead of 6). The loop in 1HML is inherently difficult to sample. Not only is it long, but there is also little empty space available for it.

[1] The algorithms are written in C++ and runs under Linux. Running times were obtained on a 3GHz Intel Pentium processor with 1GB of RAM.



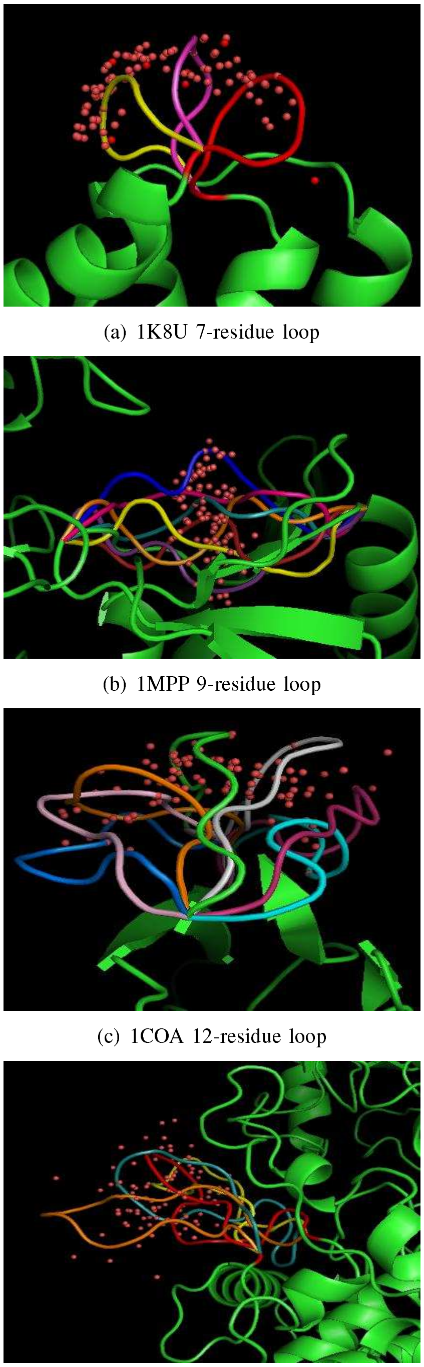
(a) 1TIB 8-residue loop

(b) 3SEB 10-residue loop

(c) 8DFR 13-residue loop

(d) 1THW 14-residue loop

Fig. 1. Some backbone conformations generated by seed sampling for the loops in 1TIB, 3SEB, 8DFR, and 1THW.



(a) 1K8U 7-residue loop

(b) 1MPP 9-residue loop

(c) 1COA 12-residue loop

(d) 1G5A 17-residue loop

Fig. 2. Positions of the middle Cα atom (red dots) in 100 loop conformations computed by seed sampling for four proteins: 1K8U, 1MPP, 1COA, and 1G5A.

See Figure 3, where the red conformation of the loop was obtained from the PDB and the other three conformations were sampled by deformation sampling. Other experiments not reported here indicate that the running times reported in Table I vary moderately when parameters like the factor ε and the number of residues in the loop's mid-portion M are slightly modified.

Figure 4 displays RMSD histograms generated for the loop in 3SEB. The purple (resp., white) histogram was obtained by sampling 100 (resp. 1000) conformations of the corresponding loop and plotting the frequency of the RMSDs between all pairs of conformations. The almost identity of the two histograms indicates that the sampled conformations spread quickly in $\mathbf{Q}^{\text{free}}_{\text{closed}}$. Similar histograms were generated for other loops.

For rather long loops, any seed sampling procedure that samples broadly $\mathbf{Q}^{\text{free}}_{\text{closed}}$ can only produce a coarse distribution of samples. Indeed, for a loop with n dihedral angles, a set of N evenly distributed conformations defines a grid with

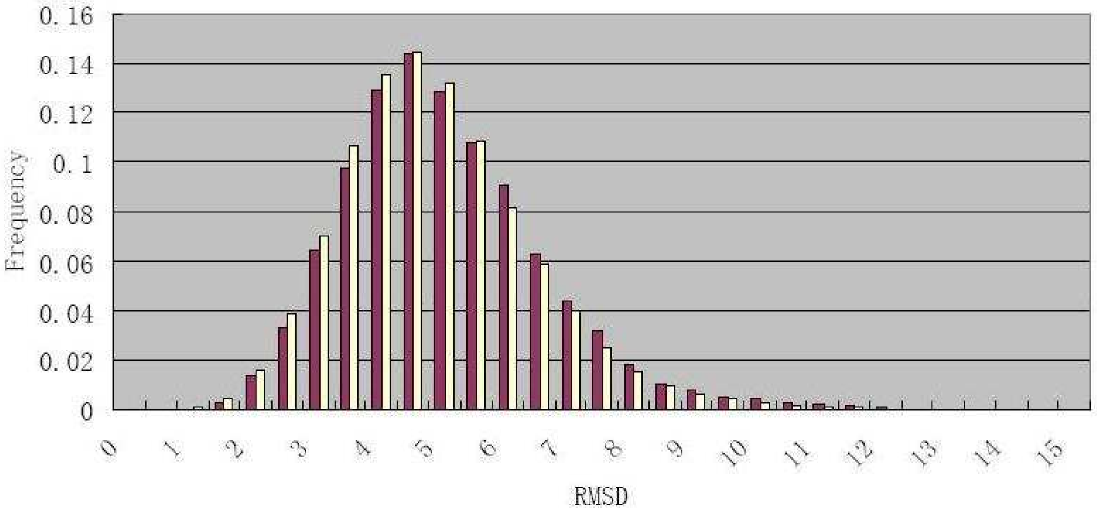


Fig. 4. RMSD histograms for one 10-residue loop in protein 3SEB. The purple color shows the pairwise RMSD distribution of 100 seeds, while the white color shows that of 1000 seeds.

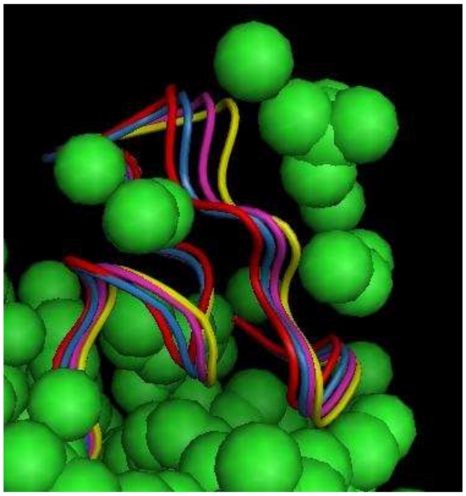


Fig. 3. Conformations of the loop in 1HML.

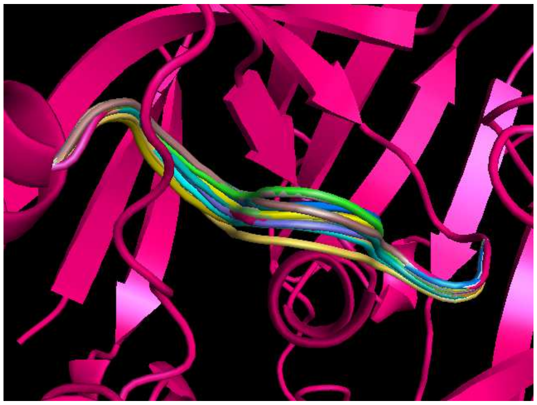


Fig. 5. Twenty conformations of the loop in 1MPP generated by deforming a given seed conformation along randomly picked directions.

$N^{1/n-6}$ discretized values for each of the $n-6$ dimensions of $\mathbf{Q}^{\text{free}}_{\text{closed}}$. If $n = 18$ (9-residue loop), a grid with 3 discretized values per axis requires sampling 531,441 conformations. Deformation sampling makes it possible to sample more densely "interesting" regions of $\mathbf{Q}^{\text{free}}_{\text{closed}}$.

B. Deformation sampling

Figure 5 shows 20 conformations of the loop in 1MPP generated by deformation sampling around a conformation computed by seed sampling. To produce each conformation, the deformation sampling procedure started from the same seed conformation and selected a short vector δq in $T\mathbf{Q}_{\text{closed}}(q)$ at random. This figure illustrates the ability of deformation sampling to explore $\mathbf{Q}^{\text{free}}_{\text{closed}}$ around a given conformation.

Figure 6 shows a series of closed clash-free conformations of the loop in 1COA successively sampled by pulling the N atom (shown as a white dot) of THR 58 away from its initial position along a given direction until a steric clash occurs (white circle). The initial conformation shown in red was generated by seed sampling and the side-chains were placed without clashes using SCWRL3. Each other conformation was sampled by deformation sampling starting at the previously sampled conformation and using the objective function E defined by Eq. (1) in Section V-C. Only the backbone was deformed, and each side-chain remained rigid. Steric clashes were tested for all atoms in the loop.

Figure 7 shows (in green) an approximation of the volume reachable by the 5th Cα atom in the loop of 1MPP. This approximation was obtained by sampling 20 seed conformations of the loop and, for each of these conformations, pulling the 5th Cα atom along several randomly picked directions until a clash occurs. The volume shown green was obtained by rendering the atom at all the positions it reached.

The running time of deformation sampling depends on the objective function. In the above experiments, it less than 0.5 seconds per sample on average.

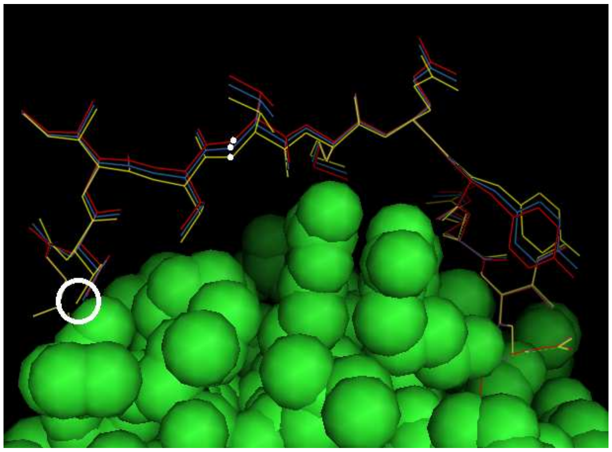


Fig. 6. Deformation of the loop in 1COA by pulling the N atom (white dot) of THR 58 along a specified direction.

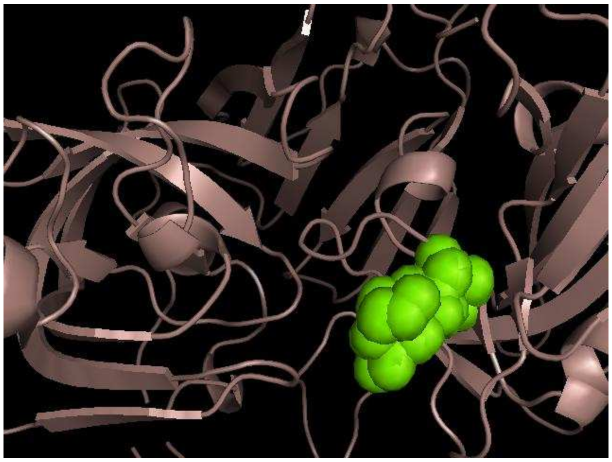
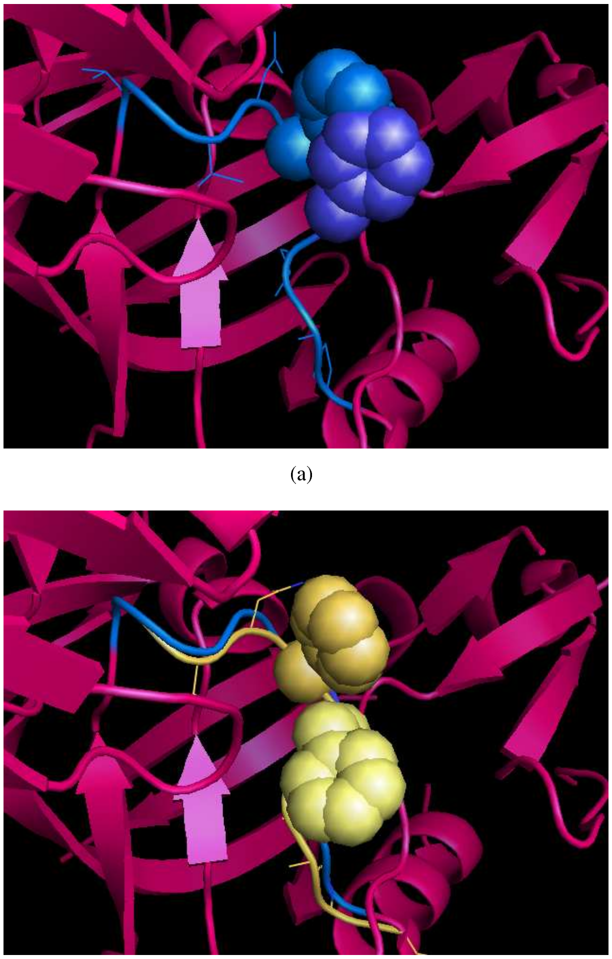


Fig. 7. Volume reachable by the 5th Cα atom in the loop of 1MPP.

C. Placements of side-chains

Our software calls SCWRL3 [5] to place side chains. The result, however, is not guaranteed to be clash-free. To generate Table II, we first ran our seed sampling procedure to sample conformations of the backbones of the loops in 1K8U, 2DRI, 1TIB, 1MPP, and 135L, with the uniform and Ramachandran sampling distributions for the dihedral angles (see Sections IV-B and IV-C). For each loop, we sampled 50 conformations with the uniform distribution and 50 with the Ramachandran distribution. We then ran SCWRL3 to place side-chains in the loop (with the side-chains in the rest of the protein fixed) and checked each conformation for steric clashes. Table II reports the number of clash-free conformations (out of 50) for each loop. As expected, the backbone conformations generated using the Ramachandran distribution facilitate the clash-free placement of the side-chains.

When seed sampling generates a conformation q of a loop backbone, such that SCWRL3 computes a side chain placement that is not clash-free, deformation sampling can then be used to sample more conformations around q, to produce one where side chains are placed without clashes. In Figure 8(a) a conformation (shown blue) of the backbone of the loop in 1MPP was generated using seed sampling and the side chains were placed by SCWRL3. However, there are clashes between two side chains. In (b) a conformation (shown yellow) was generated by the deformation sampling procedure using the conformation shown in (a) as the start conformation. The new placement of the side chains computed by SCWRL3 is free of clashes. Once such a clash-free conformation has been obtained, many other clash-free conformations can be quickly generated around it, again using deformation sampling, as shown in Figure 5.



(a)

(b)

Fig. 8. Use of deformation sampling to remove steric clashes involving side chains.

Protein	1K8U	2DRI	1TIB	1MPP	135L
Uniform	7	9	1	0	9
Ramachandran plots	18	14	6	4	13

TABLE II
NUMBER OF CLASH-FREE PLACEMENTS OF SIDE CHAINS FOR FIVE LOOPS.

D. Calcium-binding site prediction

Calcium-binding proteins play a key role in signal transduction. Many such proteins share the same functional domain, a helix-turn-helix structural motif called EF-hand [19]; the

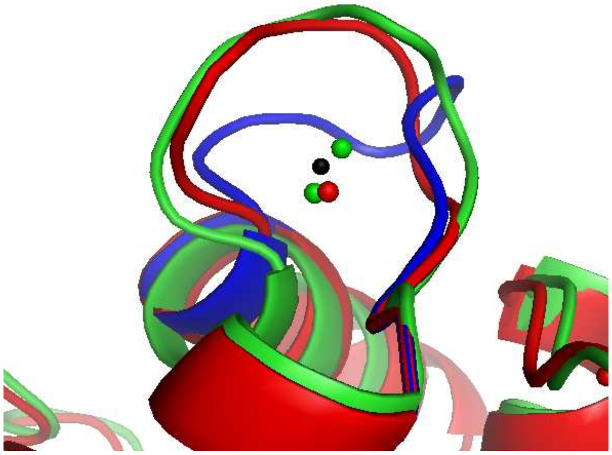


Fig. 9. Parvalbumin loop ALA51-ILE58: The apo and holo conformations recorded in the PDB are shown blue and green, respectively. The loop conformation in red is the conformation generated by seed sampling and recognized by FEATURE as a calcium-binding site. The black dot is the position of the calcium ion recorded in the PDB. The green and red dots are the calcium positions predicted by FEATURE for the loop conformations of the same color.

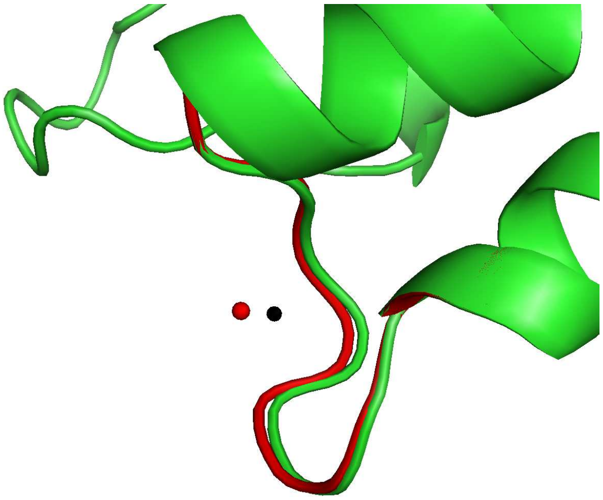


Fig. 10. Grancalcin loop ALA62-ASP69. The holo conformation in the PDB file is shown in green. The conformation in red was generated using deformation sampling FEATURE correctly recognized the red conformation as a calcium-binding site, but failed to do so on the green conformation (see text).

calcium ion binds at the loop region in this motif. As a loop is often flexible, its conformation with calcium bound (called the *holo* state) and its conformation without calcium (the *apo* state) can be significantly different [1].

Many functional site prediction methods, for example FEATURE [30], are based on structural properties of the binding site. However, if the conformation of the functional site changes upon calcium binding, these methods may not be able to recognize the binding site in the apo state due to the absence of the binding structural properties. One way to overcome this problem is to sample many closed clash-free conformations of the loop and run the functional site prediction method on each of them. If a sampled conformation is recognized by the method, not only does this indicate that the loop may be a possible calcium-binding site, it also tells us what the holo conformation may look like. In fact, molecular dynamics simulation has already been used successfully to generate conformations starting with apo proteins in order to identify unrecognized calcium binding sites in them [14].

For example, Parvalbumin [6] is a calcium-binding protein, where the loop ALA51-ILE58 is a binding site that flips up upon calcium-binding. The PDB codes for its apo and holo structures are 1B8C and 1B9A, respectively. In Figure 9, these conformations are shown blue and green, respectively; the black dot is the center of the calcium ion in the holo PDB file. We sampled successive conformations of this loop using our seed sampling procedure and ran FEATURE on each of them, until FEATURE recognized a loop conformation as a calcium-binding site. The recognized conformation, shown red in Figure 9, is close to the holo structure 1B9A. The red dot represents the position of the calcium ion predicted by FEATURE in this recognized conformation. Similarly, the two green dots represent positions of the calcium ion predicted by FEATURE for the green holo conformation. Note that all these dots are all very close to the calcium position recorded in the PDB. Correctly, FEATURE did not recognize the apo conformation shown blue as a binding-conformation; hence, there is no blue dot in the figure. We then explore the neighboring conformations of the seed, trying to get conformations even closer to the PDB holo state. We deformed the seed by deformation sampling until FEATURE returned a higher score than the seed. The final conformation only slightly improved the backbone RMSD to the holo conformation.

Deformation sampling can also be used to enhance the performance of FEATURE. To recognize a binding site, FEATURE counts atoms contained in concentric spherical shells. Therefore, it is somewhat sensitive to the values of the radii of the shells, as well as to the position of the center of the shells. This may cause FEATURE to fail to correctly recognize a functional state. For example, in protein grancalcin, the loop ALA62-ASP69 is a calcium-binding site [18]. The holo structure has PDB code 1K94. It is shown in green in Figure 10, where the black dot is the position of the calcium ion recorded in the PDB. Surprisingly, FEATURE failed to recognize this structure as a binding site. So, we then used deformation sampling around the holo structure 1K94 and ran FEATURE on each one of them until FEATURE identified it as a calcium-binding site. The resulting loop conformation is shown red in Figure 10, where the red dot is the predicted calcium position. The main difference between the holo structure 1K94 and the conformation generated by deformation sampling is the location of ASP65, one of the four coordinating residues. Atoms from the main and side chains of ASP65 are located slightly closer to the calcium binding site in the conformation obtained by deformation sampling. These small displacements are sufficient to change the atom counts in the spherical shells considered by FEATURE, thereby affecting the score of the entire site.

VIII. Conclusion

We have described two distinct algorithms to sample the space of closed clash-free conformations of a flexible loop. The seed sampling algorithm produces broadly distributed conformations. It is based on a novel prioritized constraint-satisfaction approach that interweaves the treatment of the clash avoidance and closure constraints. The deformation sampling algorithm uses seed conformations as starting points to explore more finely certain regions of the space. It is based on the computation of the null space of the loop backbone at its current conformation.

Early versions of these algorithms have been used successfully to interpret fuzzy regions in electron-density maps obtained from X-ray crystallography experiments [27]. Computational tests reported in this paper show that our algorithms can efficiently handle loops ranging from 5 to 25 residues in length. Additional tests demonstrate their ability to generate biologically interesting loop conformations, such as calcium-binding conformations. This critical ability could be used in the future to predict loop conformations and improve other structure prediction techniques, like homology, when functional information is known in advance.

Acknowledgements: This work has been partially supported by NSF grant DMS-0443939. Peggy Yao was supported by a Bio-X graduate fellowship. Russ B. Altman acknowledges support of NIH grants LM-05652 and GM072970, supporting the Simbios National Center for Physics-Based Simulation of Biological Structures.

References

[1] Babor, M., Greenblatt, H.M., Edelman, M., and Sobolev, V. Flexibility of metal binding sites in proteins on a database scale. *Proteins: Struc., Func., and Bioinf.*, **59** (2005) 221–230.

[2] Bakker, P. I. W. de, DePristo, M. A., Burke, D. F., and Blundell, T. L., Ab Initio Construction of Polypeptide Fragments: Accuracy of Loop Decoy Discrimination by an All-Atom Statistical Potential and the AMBER Force Field with the Generalized Born Solvation Model, *Proteins: Struc., Func., and Gene.* **51** (2003) 21–40.

[3] Bruccoleri, R.E. and Karplus, M. Conformational sampling using high temperature molecular dynamics. *Biopolymers* **29** (1990) 1847–1862.

[4] Canutescu, A. and Dunbrack Jr., R. Cyclic coordinate descent: A robotics algorithm for protein loop closure, *Protein Sci.* **12** (2003) 963–972.

[5] Canutescu, A., Shelenkov, A., and Dunbrack Jr., R. A graph theory algorithm for protein side-chain prediction, *Protein Sci.*,**12** (2003) 2001–2014.

[6] Cates, M.S., Berry, M.B., Ho, E.L., Li, Q., Potter, J.D., and Phillips Jr., G.N. Metal-ion affinity and specificity in EF-hand proteins: coordination geometry and domain plasticity in parvalbumin. *Structure Fold. Des.*, **7** (1999) 1269-1278.

[7] Chang, K.S. and Khatib, O. Operational space dynamics: Efficient algorithm for modeling and control of branching mechanisms. *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, (2000) pp. 850–856.

[8] Cortes, J., Simeon, T., Renaud-Simeon, M., and Tran, V. Geometric algorithms for the conformational analysis of long protein loops, *J. Comp. Chem.*, **25** (2004) 956–967.

[9] Coutsias, E.A., Soek, C., Jacobson, M.P., and Dill, K.A. A kinematic view of loop closure, *J. Comp. Chem.*, **25** (2004) 510–528.

[10] Deane C.M. and Blundell T.L. A novel exhaustive search algorithm for predicting the conformation of polypeptide segments in proteins. *Proteins: Struc., Func., and Gene.* **40** (2000) 135–144.

[11] DePristo, M.A., de Bakker, P.I.W., Lovell, S.C., and Blundell, T.L. Ab initio construction of polypeptide fragments: efficient generation of accurate, representative ensembles. *Proteins: Struc., Func., and Gene.* **51** (2003) 41–55.

[12] Fiser, A., Do R.K.G., and Sali, A. Modeling of loops in protein structures. *Protein Sci.* **9** (2000) 1753–1773.

[13] Furnham, N., Blundell, T.L., DePristo, M.A., and Terwilliger, T.C. Is one solution good enough? *Nature Structure Molecular Biology* **13** (2006) 184–185.

[14] Glazer, D.S., Radmer, R.J., and Altman, R.B. Combining Molecular Dynamics and Machine Learning to Improve Protein Function Recognition. *Pacific Symposium on Biocomputing*, **13** (2008) 332–343.

[15] Golub, G. and van Loan, C. *Matrix Computations*, John Hopkins University Press, 3rd edition, 1996.

[16] Halperin, D. and Overmars, M.H. Spheres, molecules and hidden surface removal. *Comp. Geom. Theory and App.*, **11** (1998) 83-102.

[17] Jacobson, M.P., Pincus, D.L., Rapp, C.S., Day, T.J.F., Honig, B., Shaw, D.E., and Friesner, R.A. A hierarchical approach to all-atom protein loop prediction. *Proteins: Struc., Func., and Bioinf.*, **55** (2004) 351–367.

[18] Jia, J., Borregaard, N., Lollike, K., and Cygler, M. Structure of Ca^{2+}-loaded human grancalcin. *Acta Cryst.*, **D57** (2001) 1843–1849.

[19] Kawasaki, H. and Kretsinger, R.H. Calcium-binding Proteins 1:EF-hands. *Protein Profile*, **2** (1995) 305–490.

[20] Kolodny, R., Guibas, L., Levitt, M., and Koehl, P. Inverse kinematics in biology: the protein loop closure problem. *Int. J. Robotics Research* **24** (2005) 151–163.

[21] Levin, E., Kondrashov, D., and Wesenberg, G. Ensemble refinement of protein crystal stuctures: validation and application. *Structure* **15** (2007) 1040–1052.

[22] Okazaki, K., Koga, N., Takada, S., Onuchic, J.N., and Wolynes, P.G. Multiple-basin energy landscapes for large-amplitude conformational motions of proteins: Structure-based molecular dynamics simulations. *PNAS* **103** (2006) 11844–11849.

[23] Sauder, J.M. and Dunbrack Jr., R. Beyond genomic fold assignment: rational modeling of proteins in bilological systems, *J. Mol. Biol.*, **8** (2000) 296–306.

[24] Shehu, A., Clementi, C., and Kavraki, L.E. (2006) Modeling protein conformation ensembles: From missing loops to equilibrium fluctuations, *Proteins: Structure, Function, and Bioinformatics*, **65** (2006) 164–179.

[25] Sousa, S.F., Fernandes, P.A., and Ramos, M.J. Protein-ligand docking: Current status and future challenges. *Proteins: Struc., Func., and Bioinf.*, **65** (2006) 15–26.

[26] Tossato, C.E., Bindewald, E., Hesser, J., and Manner, R. A divide and conquer approach to fast loop modeling. *Protein Eng.* **15** (2002) 279–286.

[27] van den Bedem, H., Lotan, I., Latombe, J.C., and Deacon, A. Real-space protein-model completion: an inverse-kinematic approach, *Acta Cryst.*, **D61** (2005) 2–13.

[28] van Vlijmen, H.W.T. and Karplus, M. PDB-based protein loop prediction: parameters for selection and methods for optimization. *J. Mol. Biol.* **267** (1997) 975–1001.

[29] Wedemeyer, W.J. and Scheraga, H.A. Exact analytical loop closure in proteins using polynomial equations. *J. Comp. Chem.*, **20** (1999) 819–844.

[30] Wei, L. and Altman, R.B. Recognizing protein binding sites using statistical descriptions of their 3D environments. *Pacific Symposium on Biocomputing (PSB)*, (1998) 497–508