Computing Protein Structures from Electron Density Maps: The Missing Loop Problem

Itay Lotan[1], Henry van den Bedem[2], Ashley M. Deacon[2], and Jean-Claude Latombe[1]

[1] Computer Science Department, Stanford University, Stanford CA 94305.
E-mail: itayl@cs.stanford.edu, latombe@cs.stanford.edu
[2] Joint Center for Structural Genomics, Stanford Synchrotron Radiation Laboratory, Stanford Linear Accelerator Center, 2575 Sand Hill Road, Menlo Park CA 94025
E-mail: vdbedem@slac.stanford.edu, adeacon@slac.stanford.edu

Abstract. Rapid protein structure determination relies greatly on the availability of software that can automatically generate a protein model from an experimental electron density map. Tremendous advances in this area have been achieved recently. In favorable cases, available software can build over 90% of the final model. However, in less favorable circumstances, particularly at medium-low resolution, only about 2/3 completeness is attained. Manual completion of these partial models is usually feasible but time-consuming. The electron density in areas of missing fragments is often of poorer quality, especially for flexible loops, making manual interpretation particularly difficult. Except for the beginning and end of the protein chain, the end points of each missing fragment are known from the partial model. Thanks to the kinematic chain structure of the protein backbone, loop completion can be approached as an inverse kinematics problem. A fast, two-stage inverse kinematics algorithm is presented that fits a protein chain of known sequence to the electron density map between two anchor points. Our approach first samples a large set of candidates that meet the closure constraint and then refines the most promising candidates to improve the fit. The algorithm has been tested and used to aid protein model completion in areas of poor density, closing loops of up to 12 residues to within 0.25Å RMSD of the final refined structure. It has also been used to close missing loops of the same length in partial models built at medium-low resolution to within 0.6Å.

1 Introduction

A protein is a linear sequence of amino acids (residues) that form a polypeptide chain. The function of a protein is largely dictated by its folded, 3-D structure, which determines its ability to bind to other molecules, such as small ligands, other proteins, or DNA. However, in general, predicting the folded state of a protein of a given sequence is still beyond our ability.

X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy are two widely used experimental techniques to obtain atomic coordinates of macromolecular structures. NMR spectroscopy enables structural biologists

to observe dynamic properties of proteins, but is limited in protein size. X-ray crystallography has no such limitation, but has the drawback that it requires crystallizing protein samples. Once a sequence of X-ray diffraction images is collected from a protein in crystalline form, a distribution of electronic charge in $\mathbb{R}^3$ of the atoms constituting the molecule can be calculated. Coordinates of each atom are obtained by interpreting this 3-D *electron density map* (EDM) [23]. This consists of placing the atoms in 3-D space to best match the electron density. The process, from synthesizing the protein in the lab to depositing the obtained coordinates of its folded structure in the Protein Data Bank (PDB) [5] may take several months, and sometimes years.

The Protein Structure Initiative (PSI), a National Institute of General Medical Sciences program in the US, aims to reduce the time and associated costs of determining a 3-D protein structure [43]. The long-range goal of the PSI is to produce structural coverage of a majority of sequenced genes. Tremendous advances in automated model building have been made over the past few years. Various software systems are now capable of generating a protein model from an EDM without human intervention [33,44,55,58].

The degree of completeness of these initial models, i.e., the fraction of residues correctly placed, varies with the quality of the experimental data, yet rarely reaches 100%. Mobility of some fragments of the molecule, temperature-dependent atomic vibration and other factors may lead to EDM of poor quality locally, making automatic interpretation difficult. Manually completing a partial protein model, i.e. building in the missing residues is often feasible, but is time-consuming.

In practice, a large portion of the molecule is often resolved successfully. The end points of any gaps in the initial model are known as well as the length and sequence of the missing fragment (referred to as *loop*). Since a protein molecule can be modelled as a long kinematic chain — with rigid groups of atoms as links and rotatable bonds as joints — loop fitting in this case is similar to an inverse kinematics (IK) problem [48,49]. One needs to compute values for the degrees of freedom (DoFs) of the protein that optimize its fit with the density while respecting the closure constraint — the loop must bridge the gap between the two endpoints.

Exploiting this similarity, we have developed a fast, two-stage algorithm based on IK techniques to fit a protein chain to the EDM between two anchor residues. The first stage aims to sample a large number of closing conformations[1], guided by the EDM. These candidate conformations are ranked according to density fit and conformational likelihood. Top-ranking conformations are then refined using an optimization procedure that locally searches the sub-space of closed loop conformation for the optimal loop structure.

The algorithm has been successfully tested and used to complete initial models. At a resolution of 2.0Å, it closed gaps as long as 12 residues to within

[1] The chemical equivalent term for a configuration in robotics

0.25Å RMSD[2] of the manually resolved structure. In a partial model built at 2.8Å, it closed missing loops of the same length to within 0.6Å all atom RMSD in areas of poor density.

2 Description of problem

Rapid protein structure determination depends critically on the availability of software that can automatically generate a protein model from an EDM. At high resolution, existing programs may provide over 90% of the protein main chain of the final model [4]. At medium to low resolution levels ($2.3\text{Å} \leq d < 2.9\text{Å}$), the initial model resulting from these programs is typically a gapped polypeptide chain, and only about 2/3 completeness is attained. Programs targeting lower resolution levels, notably *RESOLVE* [58] and *MAID* [44], rely on pattern recognition techniques, unambiguous density and elementary stereochemical constraints. Thus, poorly defined areas of the EDM pose a considerable challenge.

In a crystallographic experiment, the phase angle of the diffracted beam is lost, and only the magnitudes are recorded [23]. The phase angle is recovered at a later stage, but may have a substantial error associated with it, leading to systematic errors in the EDM. The resolution at which diffraction data was collected also affects the interpretability of the map. High resolution data allows the crystallographer to distinguish detail at the atomic level, whereas the EDM appears "blurred" at lower resolution. It is often difficult or even impossible to obtain high resolution EDMs of some proteins. A protein crystal contains many replicas of the protein, which all contribute to the resulting EDM. If the structures of these replicas are not identical, localized disorder in the EDM may result. Temperature-dependent atomic vibrations and the existence of mobile regions in the protein are among the causes of local disorder.

Initial protein models, whether obtained manually or by automated procedures, are only approximately correct. Their coordinates typically serve as initial values for standard maximum likelihood (ML) algorithms, which improve the model coordinates using appropriate statistical techniques [53]. Iterating model building and refinement steps to improve the completeness and quality of atomic models has clear advantages [55,59]. Phase information from the updated model is combined with experimental phases to improve the electron density that is used to generate a new model. Missing fragments in a model hinder the ability of this procedure to converge to the correct phases and model, since significant parts of the EDM remain unaccounted for by the model. Thus, filling in the gaps with fragments whose coordinates are within the radius of convergence of ML algorithms may significantly improve the EDM at an early stage in the process.

[2] Root of the sum of the squared distances between corresponding atoms. It is computed after the loops are optimally aligned in 3-D.

The input to our algorithm is the EDM, the parts of the structure that were resolved by the automatic model builder, and the two anchor residues that need to be bridged (henceforth denoted N-anchor and C-anchor). In the majority of partially resolved structures, the amino acid sequence is correctly assigned. Thus, we assume that the gap length and residue sequence of the missing fragment are known. Our goal is to propose candidate structures for the missing fragment that fall within the radius of convergence of existing refinement tools (1 - 1.5Å RMSD [53]).

The EDM is likely to have systematic errors due to erroneous phases. Consequently the parts that are solved — including the anchor residues of the missing fragment — may also be misplaced. Moreover, the density in the gap is somewhat disordered and thus not entirely reliable. Our protein model assumes *idealized geometry*, namely all bond lengths and bond angles are fixed to their idealized values [25]. These parameters vary very little in proteins [31], and their variance falls within the error margins of our input. Thus, the only DoFs in our model are the backbone torsional angles denoted the ϕ and ψ angles (one pair per residue). To simplify slightly our problem we will not have any side-chain DoFs and therefore no side-chain atoms in our protein model, with the exception of the Cβ atom. Side-chains can be built onto the model once the backbone is fully in place. We will include all Cβ and O atoms since they do not add any DoFs, yet help to orient the backbone. See Figure 1 for an illustration.

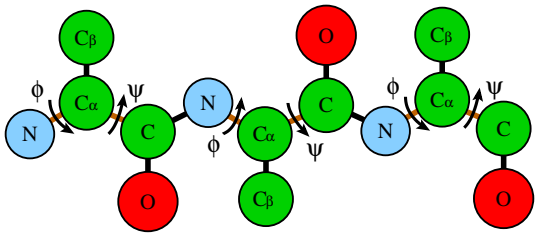


Fig. 1. A 3-residue example of the protein model used in this work

When computing the missing loop for a given gap, the two anchor residues will be incorporated onto the ends of the loop. This provides us with a measure of loop closure. Closure is measured at both ends as the RMSD distance of the three backbone atoms of the anchor residue attached to the loop (*mobile* anchor) from their positions on the anchor residue attached to the solved structure (*stationary* anchor).

3 Related Work

3.1 Exact inverse kinematics solvers

The problem of fitting a protein backbone fragment between two anchor points is closely related to the IK problem in robotics [19]. It is known that for manipulators in a 3-D workspace there are a finite number of solutions to the IK problem when the number of DoFs does not exceed six. However, there is no analytical method that can compute these solution for all types of manipulators. In the case of a 6R manipulator, which is the most relevant to this work, an analytic solution can always be obtained and the number of unique solutions is at most sixteen [56]. An efficient algorithm was derived in [47] which was later applied to the manipulation and conformational analysis of small molecular chains [48,49]. When a robot is hyper-redundant a method based on a *backbone curve* can be used (e.g., [13]). Biologists have also developed their own specialized IK tools. A method for computing conformations of ring molecules, when rotation is possible around all backbone bonds, was introduced as early as 1970 [29]. An IK solution for three consecutive residues having ideal geometry was later devised [64]. Most recently a new formulation was developed that extends the domain to any three residues (not necessarily consecutive) with arbitrary geometry [17].

3.2 IK solutions by optimization

Optimization techniques have also been applied to the IK problem. A method for bounding the exact IK solutions within small intervals in the context of drug design was proposed in [67]. When the number of DoFs exceeds six, no analytical solution is known. Currently, only optimization-based solutions exist. The *random tweak* method [27,57] closes a loop by iteratively changing its DoFs until the desired distances between its two terminals are reached. It uses the Jacobian matrix of these distances (with respect to the torsional DoFs) to compute the DoF changes at each iteration. The *cyclic coordinate descent* (CCD) [63] method closes a kinematic chain by iteratively adjusting its DoFs until a desired closure tolerance is reached. This method was later applied to protein chains [9].

3.3 Motion planning for closed loops

Searching for the loop conformation that best matches an EDM has much in common with techniques used in roadmap-based motion planning for sampling robot configurations and connecting them. A number of works offer methods for planning the motion of closed kinematic loops. Common to many of them is the use of the probabilistic roadmaps (PRM) framework [37]. The work in [66] samples configurations by first ignoring the closure constraint and then enforcing the constraint through gradient descent. Nearby configurations

are connected by the chaining of local steps that are generated in null space of the Jacobian matrix. In [32] configurations are sampled by breaking the loop into an *active* part, for which forward kinematics sampling techniques are used, and a *passive* part, for which an exact IK solution is computed. Nearby samples are connected by using a local planner for the active part and letting the passive part follow the motion. An extension of this method [16] samples the active part of the chain one DoF at a time, making sure its endpoints are always reachable by the passive part. This method was recently applied to the sampling of protein loop conformations and to the study of the flexibility of such loops [15]. Another extension to [32] is proposed in [65]. A roadmap for solving a slightly different problem, known as *point-to-point IK* for redundant robots, described in [3], also takes into account joint constraints and obstacle avoidance. A polynomial-time complete planner for the reconfiguration of closed loops with spherical joints and no obstacles is proposed in [60].

3.4 Methods for redundant manipulators

The refinement method we use exploits the redundancy of the protein chain to minimize a target function without breaking chain closure. It is closely related to the planning of instantaneous motion of redundant manipulators. Here the redundant DoFs are used for concurrent optimization of additional criteria (beyond the completion of the main task) or the execution of additional lower priority tasks. Some examples include [8,12,38,39].

3.5 The loop closure problem in biology

Our problem is closely related to the *loop closure* problem in structural biology, where loops are predicted based on sequence information alone. The methods proposed for this problem can be roughly divided into two types: *ab initio* methods and *database search* methods. The ab initio methods search the loop conformation space while database methods screen the Protein Data Bank (PDB) [5] for known structures that closely meet the requirements of the desired loop.

Ab initio methods For loops that have six DoFs or less, exact methods that enumerate all possible solutions can be used, e.g. [17,29,47,49,64]. It is also possible to use the exact solver hierarchically as in [64] and thus handle longer loops. Examples of search methods include sampling biased by the database distribution of the ϕ/ψ angle pairs [52], uniform conformational search [7], sampling from a small set of ϕ/ψ pairs [20,21] or sampling from a small library of short representative fragments [41]. Other algorithms sample conformations randomly and then enforce the closure constraints, e.g. the *random tweak* method [27,57] or the CCD-based method [9]. Various methods exist for optimization of candidate loops, such as molecular dynamics [7,28,69] and Monte Carlo [1,14] simulations.

Database search methods Work on extracting loop candidates from the PDB started with [35]. Loops are chosen that have the right length and meet the required geometric constraints. The applicability of this approach was initially limited to loops of length 4 [26], but later work suggested an upper limit of 9 [61]. A recent study claims the limit should be raised to 15 [24].

Crystallography In the crystallography community a variety of semi-automated tools and techniques have been developed to assist in completing partial models. The interactive graphics program *O* [34] selects fragments from a database to close loops. These fragments can be refined against the EDM using torsion angle refinement based on grid summation [36]. Oldfield [54] developed a method combining a random search of conformation space with grid- and gradient-based refinement techniques to close loops.

4 Methods

Our algorithm for computing missing loops proceeds in two stages: candidate generation and refinement. In the first stage, candidate loops are built using the CCD algorithm. Our implementation puts additional constraints on the DoFs to take EDM fit and collision avoidance into account. Next, initial conformations are ranked according to density fit and conformational likelihood and top-ranking ones are passed on to the refinement procedure. Refinement is achieved by minimizing a target function that quantifies the goodness of the fit of the conformation and the EDM. An optimization protocol based on simulated annealing (SA) and Monte Carlo minimization (MCM) searches for the global minimum of the target function while maintaining loop closure. Each candidate is optimized 50 times and the best scoring loops are returned.

Our approach tries to compensate for the deficient density information by taking advantage of the loop closure constraint to guide the loop to its correct positioning in space. In the first stage, the use of the closure constraint enables the generation of loops that lie within 2Å RMSD of the true solution. The approximate enforcement of the closure constraint during loop refinement prevents the search from diverging and limits the searched space to motions that preserve loop closure.

Our implementation of the algorithm discussed below, makes use of the following software packages: *Clipper* [18], the *CCP4 Coordinate Library* [42] and the exact IK solver of [17].

4.1 Stage 1: Generating loop candidates

We start by constructing a protein chain $\mathcal{C}$ of length L in a random conformation, where residue 0 is a copy of the N-anchor, and residue $L-1$ is a copy of the C-anchor. This chain is attached to either the N-, or C-anchor, thus determining the *closing direction*.

Upon starting the procedure, the position of the mobile anchor will not coincide with the position of the stationary anchor. The algorithm adjusts each backbone dihedral angle in turn such that the distance between the three backbone atoms of the mobile anchor and the corresponding atoms of the stationary anchor are minimized. A total of 2000 iterations are allowed for closure up to a preset tolerance distance d_{closed}. Chains that did not close are discarded. We calculate an average density and conformational likelihood score for all conformations, and the 95-th percentile is submitted to stage two.

Random Initial Conformations Five hundred random starting conformations are obtained from sampling $(\phi, \psi)_i, i = 0 \ldots L-1$ angle pairs from PDB-derived distributions. A finite sum of 2-D Gaussians is thereto fitted to frequencies calculated from the Top500 database [46] of non-redundant protein structures, using the program EMMIX [50]. We obtained distributions for each of the 20 amino acids. The angles ϕ_0 and ψ_{L-1} remain fixed at their initial values.

Electron Density Constraints A change to the DoFs of a residue is calculated as follows: The CCD algorithm proposes a distance minimizing dihedral angle ϕ_i for residue i. Based on ϕ_i, it will propose a minimizing ψ_i. (In our implementation, we change each DoF in turn, although this is not strictly necessary.) We will denote these angles as a proposed angle pair $(\phi, \psi)_i^p$.

To guide the loop, a heuristic electron density constraint has been added to the CCD algorithm. For each pair i, we define a set of atoms $\mathcal{A}_i$ that is subject to change by angle pair i, and not affected by changes in angle pair $i+1$. Hence, $\mathcal{A}_i = \{C\beta^i, C^i, O^i, N^{i+1}, C\alpha^{i+1}\}$.

We evaluate electron density values corresponding to trial positions in a two dimensional, square neighborhood $U_{(\phi,\psi)^p}$ about $(\phi, \psi)^p$ in conformation space. A simple local scoring function is adopted; the sum of the electron density values at atom center positions of $\mathcal{A}_i$. We set $(\phi, \psi)_i$ to the trial position with maximum density score, i.e $(\phi, \psi)_i = \arg\max_{(\phi,\psi)\in U_{(\phi,\psi)_i^p}} S(\phi, \psi)$, where $S(\phi, \psi) = \sum_{A_j \in \mathcal{A}_i} \rho(A_j(c))$, and $A_j(c)$ denotes the center of atom A_j. The size of $U_{(\phi,\psi)^p}$ is reduced linearly in the number of CCD iterations to allow closure of the chain.

4.2 Stage 2: Refining loop candidates

Target function A candidate loop structure is refined by minimizing the least squares residuals between the EDM ρ^o and the density calculated from the model ρ^c. This is a standard procedure used in crystallography for refinement of protein models [11,22]. The target function sums the squared differences between the observed density and the calculated density at each

grid point in some volume V around the loop:

$$T(q) = \sum_{g_i \in V} \left[S\rho^o(g_i) + k - \rho^c(g_i) \right]^2 . \quad (1)$$

The calculated density at each grid point is a sum of contributions of all atoms whose center lies within a cutoff distance from this point. The calculated density contribution of an atom is a sum of isotropic 3-D Gaussians [62]. The factors S and k scale ρ^o to ρ^c and are computed once at initialization.

Optimization with closure constraints Our method is based on the approach described in [8,38] for optimizing an objective function while performing a given task by taking advantage of manipulator redundancy. The redundant DoFs define a subspace of configuration space termed the *self-motion* manifold. Motions on this manifold do not influence the main task and thus can be used to move the manipulator configuration towards a minimum of the objective function. However, since this manifold may be very complex these motions are in general difficult to compute. It is therefore common to approximate the self-motion manifold by its tangent space. This space is simply defined as the null-space of the linearized instantaneous kinematic relation between the linear and angular velocities of the end effector (a frame attached to the end of the chain) and the joint velocities, also known as the chain's Jacobian matrix [19]. For an n-DoF chain in $\mathbb{R}^3$ at configuration q, the Jacobian $J(q)$ is a $6 \times n$ matrix satisfying the equation:

$$\dot{x} = J(q)\dot{q}. \quad (2)$$

Thus $J(q) = df(q)/d(q)$ where $f(q)$ is the chain's forward kinematics function mapping joint coordinates to end effector position and orientation. The rank of the Jacobian in $\mathbb{R}^3$ is at most 6 and thus the dimensionality of its null space is at least $n - 6$. In general, an instantaneous change in the configuration is computed by inverting Equation 2 and exploiting the null space to optimize the objective function. We get:

$$\dot{q} = J^{\dagger}(q)\dot{x} + N(q)N^T(q)y, \quad (3)$$

where $J^{\dagger}$ is the pseudo-inverse of the Jacobian and $N(q)$ is an orthonormal basis for the null-space. When optimizing an objective function for a closed loop, the instantaneous change in position and orientation of the end, $\dot{x}$, is set to zero. y is taken to be the gradient vector of the objective function. Projecting it onto the null space of the Jacobian produces a motion that minimizes the objective function while not disturbing the chain closure. In general, the step size of the motions (the magnitude of the vector $\dot{q}$) needs to be small since the Jacobian approximation of the self-motion manifold is valid only in a small neighborhood in configuration space surrounding the current configuration.

Implementation details

Computing the null space: We use the *atom group local frames* method [68] to represent the protein as a kinematic chain and the method prescribed in [10] to efficiently compute the Jacobian for any given sub-chain of the loop. The null space is computed by applying the SVD [30] to the Jacobian matrix. We get the decomposition $J(q) = U\Sigma V^T$, where the orthogonal matrix V is a basis of for the row-space of $J(q)$. The null-space basis $N(q)$ is simply the set of vectors of V that correspond to singular values (the diagonal elements of Σ) that are equal to zero.

Target function gradient: We derive an analytical expression for the gradient of the target function with respect to the torsional DoFs of the loop. The gradient is computed using a recursive method [2], which is linear in the number of DoFs of the chain. The naïve approach in this case has quadratic complexity.

MCM and SA: A gradient descent search for the minimum of the target function is prone to get stuck quickly in a local minimum. To increase the radius of convergence, we use the MCM approach [45]. At each step, a random move in conformation space is proposed, the new conformation is then minimized by gradient descent and the resulting local minimum is accepted or rejected using the Metropolis criterion [51]. The effect of minimization is to considerably increase the probability of acceptance of a trial move, which enables the search to make more progress. This, however, comes at the cost of significantly increasing the computation time needed for each simulation step. To further improve the ability of our search to escape from local minima we envelope the MCM method with an SA [40] protocol, which changes the pseudo-temperature of the search and thus controls the probability of acceptance of uphill moves.

Random moves: We employ two methods for generating the random moves that are needed for MCM. The first is to choose a random direction in the null-space of a randomly chosen sub-chain, and taking a step with a randomly picked magnitude that depends on the current temperature of the SA protocol. The number of DoFs in the sub-chain is at least eight so that the null space is at least two-dimensional. A similar method for generating motion on the self-motion manifold was described in [66]. Before performing minimization, we make sure the loop closure tolerance has not been exceeded, by computing the effect of the change on the end of the loop. A second method for generating random steps is the use of an exact IK solver [17]. One of the solutions is chosen at random as the proposed move. The use of an exact solver allows us to jump between unconnected parts of the self-motion manifold which the Jacobian-based moves are unable to do.

Approximate closure The closure constraint is relaxed a little during this stage. A maximum RMSD of 0.5Å is allowed at both ends of the loop. Since there are errors in the positions of the anchor residues, and because the resulting loop will be subsequently refined together with the rest of the protein, this tolerance is acceptable. By allowing approximate closure, larger steps, can be taken in the null space of the Jacobian.

Refinement search protocol The refinement protocol is composed of three nested loops (See Figure 2. The inner loop performs MCM search by using the two methods described above for generating random trial moves. The middle loop performs the SA protocol by gradually reducing the pseudo-temperature of the MCM search. The outer loop enhances the SA protocol by simulating restarts of the SA protocol each time at a lower starting pseudo-temperature. The magnitude of attempted random null-space moves is reduced together with the current pseudo-temperature of the simulation to increase the chance that the random moves will be accepted. The use of exact IK moves is discontinued below some temperature since initial tests showed that their probability of acceptance becomes practically zero below this value. They do however play an important role in the initial stages of the search, when large moves are taken. Our initial tests indicated that using exact IK moves in the early stages of the search significantly increases the chance of finally converging to the global maximum.

```
for start_temp = high_start_TEMP downto low_start_TEMP {
    temp = start_temp;
    for SA_steps = 1 to 8 {
        for MCM_steps = 1 to NUM_ITERS {
            M = ProposeRandomMove(temp);
            MinimizeMove(M);
            AcceptMove(M);
        }
        temp = temp - TEMP_step;
    }
}
```

Fig. 2. Pseudo-code for refinement search protocol

5 Experimental results

The algorithm was developed and tested on missing loops from various structures provided by the JCSG. Below, we provide results for two representative examples, protein structures TM0423 [6] (376 residues) and TM0813 (342 residues). Both are publicly available from the PDB under ID numbers 1KQ3 and 1J5X respectively.

TM0423 is annotated as a glycerol dehydrogenase from the organism Thermotoga Maritima. Standard, publicly available crystallographic software was used to obtain a data set. We built an initial model using RESOLVE at 2.0Å. 88% of the residues in this protein were correctly built and assigned to the sequence.

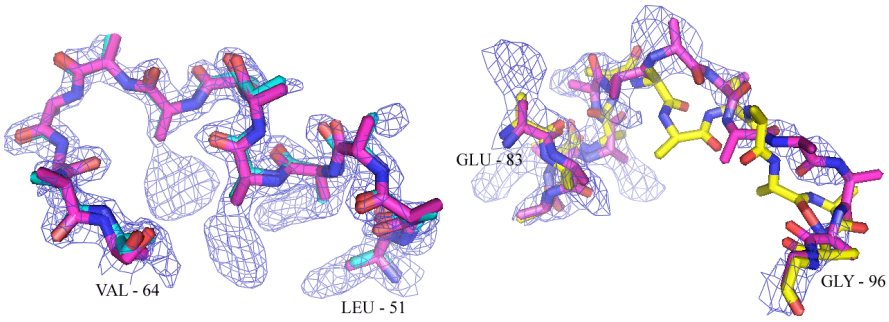
TM0813 is annotated as a conserved hypothetical protein from Thermotoga Maritima. High resolution data was excluded from the data set, and an initial model was built by RESOLVE at 2.8Å. The completeness of this model was found to be 61%.

We use RMSD between top ranking loops and their corresponding regions from the final, refined structure as a measure of performance. Note that since we use idealized geometry while the bond lengths and angles are allowed to vary in the final structure, a discrepancy of 0.2 - 0.3Å is expected even for an exactly correct loop. In refinement, the protein structure may undergo rigid translations or rotations. More importantly, the stationary anchors can move independently. To exclude the effects of refinement steps, we tested the algorithm on an artificially created gap of length 7 in TM0423. We removed residues 19 to 25 from the initial model, and compared our top ranking candidates to this extracted fragment. We found a lowest RMSD of 0.35Å.

Next, we selected gaps from the initial models. TM0423 has a loop of length 7 missing, extending from residue 251 to residue 259, and a longer loop of length 12, from residue 51 to 64. The minimum, all-atom RMSD achieved for both loops is 0.25. Figure 3a depicts the best scoring 12 residue loop together with the final, refined structure.

The low resolution model TM0813 is very incomplete, and we selected a loop of length 12 extending from residue 83 to 96. The all-atom RMSD of candidate conformations submitted to the second stage of the algorithm did not get below 2.12Å. Nonetheless, the minimum, all atom RMSD achieved for these loops upon completion of the algorithm is 0.56Å. The best scoring final conformation and its starting conformation for stage two are shown in Figure 3b.

The flipped residue problem A recurring problem in some of the loop candidates we generate is commonly known as the *flipped residue* problem. It is notoriously hard to resolve for model refinement tools. A loop candidate is generated where the backbone is in good agreement with the density, yet when comparing with the manually solved structure, the orientation of one or two residues is off by 180°. This is illustrated in Figure 4. Our elaborate SA protocol is designed to alleviate this problem but is often not enough. It is very difficult for the search algorithm to recover from such flips. It requires a large concerted move in configuration space of the DoFs in the vicinity of the flip that has a small effect on the position of the chain in the 3-D workspace. Developing heuristics to fix a flip once the refinement is completed requires



(a) (b)

Fig. 3. Fitting loops into the density. (a) Highest scoring loop for residues 51 to 64 from TM0423 in cyan, together with the PDB structure in magenta. The RMSD is 0.25A. (b) Highest scoring loop for residues 83 to 96 from TM0813 in magenta, together with its starting conformation in yellow (output of stage 1). The RMSD between the starting conformation and the PDB structure is 2.1A. The refinement procedure reduces it to 0.6A. The EDM is represented by an iso-surface shown in blue wire mesh.

knowledge of its exact location, which may be difficult to ascertain without knowing the global minimum.

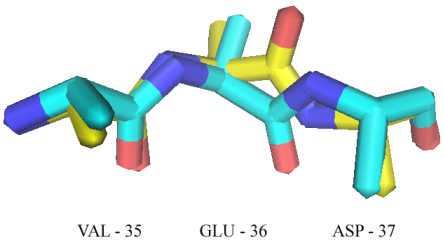


Fig. 4. An example of a one residue flip. The oxygen (red protrusion) of the middle residue in the computed loop (cyan) is in the opposite direction of the oxygen in the PDB structure (yellow) while the residues before and after are well aligned.

6 Conclusion

Existing crystallographic software sometimes fails to model parts of a protein, resulting in an initial structure with gaps. In this paper we presented a novel method for the computation of these missing fragments. The use of robotics IK techniques enabled the inclusion of a closure constraint, thus augmenting

reduced information available in areas of poor quality density. The experimental results demonstrated that our approach computed loops of up to 12 residues in good agreement with the final, refined structure. The method also performs well when using low resolution EDMs.

We have shown that protein model building procedures can greatly benefit from the use of robotics techniques and algorithms. Collaborative initiatives between the crystallography and robotics communities could therefore favorably impact the development of fully automated structure determination pipelines.

The algorithm is currently undergoing extensive testing to gauge its performance, and will eventually be made publicly available. It is anticipated that the flipped residues problem can be overcome with elementary heuristic techniques. Another interesting extension to the algorithm would be the inclusion of stereochemical constraints.

Acknowledgements: Test structures TM0423 and TM0813 used in this work were solved and deposited as part of the JCSG pipeline (www.jcsg.org). The JCSG is funded by the Protein Structure Initiative of the National Institutes of Health, National Institute of General Medical Sciences. SSRL operations is funded by DOE BES, and the SSRL Structural Molecular Biology program by DOE BER, NIH NCRR BTP and NIH NIGMS. Itay Lotan is supported in part by a Siebel Fellowship. Itay Lotan and Jean-Claude Latombe are also funded by NSF ITR grant CCR-0086013 and a Stanford BioX Research Initiative grant.

References

1. R. Abagyan and M. Totrov, *Biased probability monte carlo conformational searches and electrostatic calculations for peptides and proteins*, J. Mol. Biol. **235** (1994), 983–1002.
2. A. Abe, W. Braun, T. Noguti, and N. Gō, *Rapid calculation of first and second derivatives of conformational energy with respect to dihedral angles for proteins. general recurrent equations*, Comput. Chem. **8** (1984), no. 4, 239–247.
3. J.M. Ahuactzin and K.K. Gupta, *The kinematic roadmap: a motion planning based global approach for inverse kinematics of redundant robots*, IEEE Trans. Robot. Autom. **15** (1999), no. 4, 653–669.
4. J. Badger, *An evaluation of automated model-building procedures for protein crystallography*, Acta Cryst. **D59** (2003), 823–827.
5. H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, and et al, *The protein data bank*, Nucl. Acids Res. **28** (2000), 235–242.
6. L. S. Brinen, J. M. Canaves, X. P. Dai, A. M. Deacon, M. A. Elsliger, S. Eshaghi, R. Floyd, A. Godzik, C. Grittini, S. K. Grzechnik, and et. al., *Crystal structure of a zinc-containing glycerol dehydrogenase (tm0423) from thermotoga maritima at 1.5 angstrom resolution*, Proteins **50** (2003), no. 2, 371–374.
7. R. Bruccoleri and M. Karplus, *Prediction of the folding of short polypeptide segments by uniform conformational sampling*, Biopolymers **26** (1987), 137–168.
8. J.W. Burdick, *On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds*, IEEE Int. Conf. Robot. Autom. (ICRA), vol. 1, May 1989, pp. 264–270.
9. A.A. Canutescu and R.L. Dunbrack Jr, *Cyclic coordinate descent: A robotics algorithm for protein loop closure.*, Prot. Sci. **12** (2003), 963–972.
10. K.-S. Chang and O. Khatib, *oprerational space dynamics: efficient algorithms for modeling and control of branching mechanisms*, IEEE Int. Conf. Robot. Autom. (ICRA) (San Francisco), April 2000, pp. 850–856.
11. M.S. Chapman, *Restrained real-space macromolecular atomic refinement using a new resolution-dependent electron-density function*, Acta Cryst. **A51** (1995), no. 1, 69–80.

12. I.D. Chen, Y.-C.; Walker, *A consistent null-space based approach to inverse kinematics of redundant robots*, IEEE Int. Conf. Robot. Autom, vol. 3, May 1993, pp. 374–381.
13. G.S. Chirikjian, *General methods for computing hyper-redundant manipulator inverse kinematics*, IEEE/RSJ Int. Conf. Intel. Robots and Sys. (Yokohama, Japan), July 1993.
14. V. Collura, J. Higo, and J. Garnier, *Modeling of protein loops by simulated annealing*, Prot. Sci. **2** (1993), 1502–1510.
15. J. Cortés, T. Siméon, M. Remaud-Siméon, and V. Tran, *Geometric algorithms for the conformational analysis of long protein loops*, J. Comp. Chem., To appear.
16. J. Cortés, T. Simóon, and J.-P. Laumond, *A random loop generator for planning the motions of closed kinematic chains using prm methods*, IEEE Int. Conf. Robot. Autom. (ICRA), vol. 2, May 2002, pp. 2141–2146.
17. E.A. Coutsias, C Seok, M.P. Jacobson, and K.A. Dill, *A kinematic view of loop closure*, J. Comput. Chem. **25** (2004), 510–528.
18. K. D. Cowtan, *Clipper libraries*, 2004, http://www.ysbl.york.ac.uk/ cowtan/clipper/clipper.html.
19. J.J. Craig, *Introduction to robotics: manipulation nad control*, 2nd ed., Addison-Wesley, 1989.
20. C.M. Deane and T.L. Blundell, *A novel exhaustive search algorithm for predicting the conformation of polypeptide segments in proteins*, Proteins **40** (2000), no. 1, 135–144.
21. M.A. DePristo, P.I. de Bakker, S.C. Lovell, and T.L. Blundell, *Ab initio construction of polypeptide fragments: efficient generation of accurate, representative ensembles.*, Proteins **51(1)** (2003), 41–55.
22. R. Diamond, *A real-space refinement procedure for proteins*, Acta Cryst. **A27** (1971), no. 5, 436–452.
23. J. Drenth, *Principles of protein x-ray crystallography*, 2nd ed., Springer Verlag, New York, 1999.
24. P. Du, M Andrec, and R.M. Levy, *Have we seen all structures corresponding to short protein fragments in the protein data bank? an update*, Prot. Engin. **16** (2003), no. 6, 407–414.
25. R.A. Engh and R. Huber, *Accurate bond and angle parameters for x-ray protein structure refinement.*, Acta Cryst. **A47** (1991), 392–400.
26. K. Fidelis, P.S. Stern, D. Bacon, and J. Moult, *Comparison of systematic search and database methods for constructing segments of protein structure*, Prot. Engin. **7** (1994), no. 8, 953–960.
27. R.M. Fine, H. Wang, P.S. Shenkin, D.L. Yarmush, and C. Levinthal, *Predicting antibody hypervariable loop conformations. ii minimization and molecular dynamics studies of mcp603 from many randomly generated loop conformations*, Proteins **1** (1986), 342–362.
28. A. Fiser, R.K. Do, and A. Sali, *Modeling of loops in protein structures.*, Prot. Sci. **9(9)** (2000), 1753–73.
29. N. Gō and H.A. Scheraga, *Ring closure and local conformational deformations of chain molecules.*, Macromolecules **3** (1970), 178–186.
30. G. Golub and C. Van Loan, *Matrix computations*, 3rd ed., Johns Hopkins University Press, 1996.
31. A. Grosberg and A. Khokhlov, *Statistical physics of macromolecules*, AIP Press, New York, 1994.
32. L. Han and N.M. Amato, *A kinematics-based probabilistic roadmap method for closed chain systems*, Workshop Algo. Found. Robot. (WAFR) (B.R. Donald, K. Lynch, and D. Rus, eds.), March 2000, pp. 233–246.
33. T.R. Ioerger and J.C. Sacchettini, *The textal system: Artificial intelligence techniques for automated protein model building.*, Methods in Enzymology. (San Diego), vol. 374, Academic Press, 2003, pp. 244–270.
34. T.A. Jones and M. Kjeldgaard, *Electron-density map interpretation*, Methods in Enzymology (San Diego), vol. 277, Academic Press, 1997, pp. 173–230.
35. T.A. Jones and S. Thirup, *Using known substructures in protein model-building and crystallography*, EMBO J. **5** (1986), 819–822.
36. T.A. Jones, J.-Y. Zou, and S.W. Cowtan, *Ring closure and local conformational deformations of chain molecules.*, Acta Cryst. **A47** (1991), 110–119.
37. L.E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*, IEEE Trans. Robot. Autom. **12** (1996), no. 4, 566–580.
38. O. Khatib, *A unified approach for motion and force control of robot manipulators: The operational space formulation*, Int. J. Robot. Autom. **RA-3** (1987), no. 1, 43–53.
39. O. Khatib, L. Sentis, J.-H. Park, and J. Warren, *Whole body dynamic behavior and control of human-like robots*, International Journal of Humanoid Robotics (2004), in press.
40. S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, *Optimization by simulated annealing*, Science **220** (1983), no. 4598, 671–680.
41. R. Kolodny, L. Guibas, M. Levitt, and P. Koehl, *Inverse kinematics in biology: the protein loop closure problem*, Submitted to IJRR, 2004.

42. E. Krissinel, *Ccp4 coordinate library project*, 2004, http://www.ebi.ac.uk/ keb/cldoc/.
43. S. A. Lesley, P. Kuhn, A. Godzik, A. M. Deacon, I. Mathews, A. Kreusch, G. Spraggon, H. E. Klock, D. McMullan, T. Shin, and et. al., *Structural genomics of the thermotoga maritima proteome implemented in a high-throughput structure determination pipeline*, Proc. Nat. Acad. Sci. **99** (2002), no. 18, 11664–11669.
44. D.G. Levitt, *A new software routine that automates the fitting of protein x-ray crystallographic electron-density maps.*, Acta Cryst. **D57** (2001), 1013–1019.
45. Z. Li and H. Scheraga, *Monte Carlo-minimization approach to the multiple-minima problem in protein folding*, Proc. Natl. Acad. Sci. **84** (1987), no. 19, 6611–6615.
46. S.C Lovell, I.W. Davis, W.B. Arendall III, P.I.W de Bakker, J.M. Word, M.G. Prisant, J.S. Richardson, and D.C. Richardson, *Structure validation by Cα geometry: ϕ, ψ and Cβ deviation*, Proteins **50** (2003), no. 3, 437–450.
47. D. Manocha and J. canny, *Efficient inverse kinematics for general 6R manipulator*, IEEE Trans. Robot. Autom. **10** (1994), no. 5, 648–657.
48. D. Manocha and Y. Zhu, *Kinematic manipulation of molecular chains subject to rigid constraints.*, Proc. Int. Conf. Intell. Syst. Mol. Biol. **2** (1994), 285–293.
49. D. Manocha, Y. Zhu, and W. Wright, *Conformational analysis of molecular chains using nano-kinematics*, Comput. Appl. Biosci. **11** (1995), no. 1, 71–86.
50. G.J. McLachlan, D. Peel, K.E. Basford, and P. Adams, *The emmix software for the fitting of mixtures of normal and t-components*, J. Stat. Software **4** (1999), no. 2.
51. N. Metropolis, A. Rosenbluth, M Rosenbluth, A. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, J. Chem. Phys. **21** (1953), 1087–1092.
52. J. Moult and M.N.G. James, *An algorithm for determining the conformation of polypeptide segments in protein by systematic search*, Proteins **1** (1986), 146–163.
53. G. N. Murshudov, A. A. Vagin, and E. J. Dodson, *Refinement of macromolecular structures by the maximum-likelihood method*, Acta Cryst. **D53** (1997), 240–255.
54. T.J. Oldfield, *A number of real-space torsion-angle refinement techniques for proteins, nucleic acids, ligands and solvent.*, Acta Cryst. **D57** (2001), 82–94.
55. A. Perrakis, T.K. Sixma, K.S. Wilson, and V.S. Lamzin, *wARP: Improvement and extension of crystallographic phases by weighted averaging of multiple-refined dummy atomic models.*, Acta Cryst. **D53** (1997), 448–455.
56. M. Raghavan and B. Roth, *Kinematic analysis of the 6r manipulator of general geometry*, Int. Symp. Robot. Res. (Tokyo), 1989, pp. 314–320.
57. P.S. Shenkin, D.L. Yarmush, R.M. Fine, H.J. Wang, and C. Levinthal, *Predicting antibody hypervariable loop conformation. 1. ensembles of random conformations for ring-like structure*, Biopolymers **26** (1987), 20532085.
58. T.C. Terwilliger, *Automated main-chain model-building by template-matching and iterative fragment extension.*, Acta Cryst. **D59** (2002), 34–44.
59. ———, *Improving macromolecular atomic models at moderate resolution by automated iterative model building, statistical density modification and refinement.*, Acta Cryst. **D59** (2003), 1174–1182.
60. J.C. Trinkle and R.J. Milgram, *Complete path planning for closed kinematic chains with spherical joints*, Int. J. Robot. Res. **21** (2002), no. 9, 773–789.
61. H.W.T. van Vlijmen and M. Karplus, *PDB-based protein loop prediction: parameters for selection and methods for optimization*, J. Mol. Biol. **267** (1997), no. 4, 975–1001.
62. D. Waasmaier and A. Kirfel, *New analytical scattering-factor functions for free atoms and ions*, Acta Cryst. **A51** (1995), no. 3, 416–431.
63. L.T. Wang and C.C. Chen, *A combined optimization method for solving the inverse kinematics problem of mechanical manipulators.*, IEEE Trans. Robot. Autom. **7** (1991), 489–499.
64. W.J. Wedemeyer and H.A. Scheraga, *Exact analytical loop closure in proteins using polynomial equations.*, J. Comput. Chem. **20** (1999), 819–844.
65. D. Xie and N.M. Amato, *Kinematics-based probabilistic roadmap method for high dof closed chain systems*, IEEE Int. Conf. Robot. Autom. (ICRA), 2004, To appear.
66. J. Yakey, LaValle S.M., and L.E. Kavraki, *Randomized path planning for linkages with closed kinematics chains*, IEEE Trans. Robot. Autom. **17** (2001), no. 6, 951–959.
67. M. Zhang and L.E. Kavraki, *Finding solutions of the inverse kinematics problems in computer-aided drug design*, Tech. Report TR02-385, Rice University, 2002.
68. ———, *A new method for fast and accurate derivation of molecular conformations*, J. Chem. Info. Comp. Sci. **42** (2002), no. 1, 64–70.
69. Q. Zheng, R. Rosenfeld, S. Vajda, and C. DeLisi, *Loop closure via bond closure and relaxation*, J. Comput. Chem **14** (1992), 556–565.