Cyclic coordinate descent: A robotics algorithm for protein loop closure

Adrian A. Canutescu and Roland L. Dunbrack, Jr.

Protein Sci. 2003 12: 963-972
Access the most recent version at doi:10.1110/ps.0242703

Cyclic coordinate descent: A robotics algorithm for protein loop closure

ADRIAN A. CANUTESCU AND ROLAND L. DUNBRACK JR.
Institute for Cancer Research, Fox Chase Cancer Center, Philadelphia, Pennsylvania 19111, USA
(RECEIVED December 13, 2002; FINAL REVISION February 17, 2003; ACCEPTED February 18, 2003)

Abstract

In protein structure prediction, it is often the case that a protein segment must be adjusted to connect two fixed segments. This occurs during loop structure prediction in homology modeling as well as in ab initio structure prediction. Several algorithms for this purpose are based on the inverse Jacobian of the distance constraints with respect to dihedral angle degrees of freedom. These algorithms are sometimes unstable and fail to converge. We present an algorithm developed originally for inverse kinematics applications in robotics. In robotics, an end effector in the form of a robot hand must reach for an object in space by altering adjustable joint angles and arm lengths. In loop prediction, dihedral angles must be adjusted to move the C-terminal residue of a segment to superimpose on a fixed anchor residue in the protein structure. The algorithm, referred to as cyclic coordinate descent or CCD, involves adjusting one dihedral angle at a time to minimize the sum of the squared distances between three backbone atoms of the moving C-terminal anchor and the corresponding atoms in the fixed C-terminal anchor. The result is an equation in one variable for the proposed change in each dihedral. The algorithm proceeds iteratively through all of the adjustable dihedral angles from the N-terminal to the C-terminal end of the loop. CCD is suitable as a component of loop prediction methods that generate large numbers of trial structures. It succeeds in closing loops in a large test set 99.79% of the time, and fails occasionally only for short, highly extended loops. It is very fast, closing loops of length 8 in 0.037 sec on average.

Keywords: Homology modeling; loop modeling; protein structure prediction; inverse kinematics; robotics; cyclic coordinate descent; loop closure

To characterize biological processes both in physiological and pathological conditions, knowledge of the three-dimensional structures of the proteins involved is of great importance. The number of unique sequences in the Protein Data Bank (PDB; Berman et al. 2000) of experimentally determined structures is now >12,000, and the number of sequences in the nonredundant protein sequence database is >1.2 million (Wheeler et al. 2002). Homology modeling remains the most accurate structure prediction method for bridging the gap between the number of sequences and available structures. At least one-third of protein sequences in most genomes are homologous at least in part to proteins in the PDB (Sauder and Dunbrack Jr. 2000), and are therefore candidates for homology modeling. Ab initio folding simulations have also made gains in recent years as the necessary computational resources have become cheaper and more plentiful (Simons et al. 1999). For domains without representatives in the PDB, these methods may provide at least a preliminary model that can be tested experimentally.

Homology modeling usually proceeds via a number of steps: (1) identification of a homolog of known structure (the "parent") for the sequence of interest (the "target"); (2) refinement of the target–parent sequence alignment through application of varied alignment methods or manual adjustment in light of the known structure; (3) backbone modeling by borrowing of core secondary structures and loops of conserved length from the parent structure, and loop mod-

Reprint requests to: Roland L. Dunbrack Jr., Institute for Cancer Research, Fox Chase Cancer Center, 7701 Burholme Avenue, Philadelphia, PA 19111, USA; e-mail: RL_Dunbrack@fccc.edu; fax: (215) 728-2412.
Article and publication are at http://www.proteinscience.org/cgi/doi/10.1110/ps.0242703.

eling in regions of the alignment that contain insertions and deletions, or in which the sequence has diverged substantially; (4) side-chain modeling onto the backbone given the target sequence; (5) refinement and validation of the model. In practice, backbone modeling and side-chain modeling are interdependent. Some methods proceed by defining constraints from the known structure and the target–parent sequence alignment, rather than borrowing Cartesian coordinates directly (Sali and Blundell 1993).

Although improvements in identification (Karplus et al. 1998; Jones et al. 1999), alignment quality (Sauder et al. 2000) and side-chain prediction methods (Bower et al. 1997; Dunbrack Jr. 1999; Mendes et al. 2001; Xiang and Honig 2001; Liang and Grishin 2002) have been significant in recent years, loop modeling remains a difficult task for several reasons. For long loops, the number of available conformations is enormous, and rapid and thorough sampling is a challenge. Defining an energy function that can select the right loop structure in an environment that is only approximately correct for the target sequence is also quite difficult. Most loop methods have not been tested in true homology modeling situations, but rather in the more artificial situation of replacing a loop back into its own native structure.

Modeling a loop requires satisfying the constraint of connecting the two protein segments on either end of the loop with a physically reasonable peptide conformation. The fixed residues on either side of the loop to be modeled are termed the N- and C-terminal anchor residues. The anchors place a constraint on the available conformations, thus reducing the size of the conformational space, but satisfying the constraint presents an algorithmic challenge. The "loop closure" problem arises in nearly all loop-modeling methods, regardless of their nature. This is the case whether loop closure takes place in the context of homology modeling or in ab initio protein structure prediction, in which, for instance, secondary structures may be moved as a whole and new loops constructed to connect them. Database methods (van Vlijmen and Karplus 1997) that borrow loops from unrelated structures that approximately fit the anchors must refine the loop structure to fit the actual anchors of the target model. Ab initio loop-modeling methods (Bruccoleri and Karplus 1987) may generate large numbers of random conformations. If these are built starting from the N-terminal anchor in the model, then the loop must be adjusted to connect the C-terminal residue of the loop to the C-terminal anchor of the model. Some methods build randomly from both the N- and C-terminal anchors, and the resulting segments must be connected in the middle (Moult and James 1986).

Several solutions have been presented to solve the "loop closure" problem. Wedemeyer and Scheraga (1999) have solved the problem analytically for tripeptides with 6 degrees of freedom. Shenkin et al. described an algorithm based on the Jacobian matrix of first derivatives of distances between atoms of the terminal residues of the loop with respect to the dihedral degrees of freedom (Fine et al. 1986; Shenkin et al. 1987). Their method, referred to as "random tweak," uses Lagrange multipliers to minimize changes in the dihedral angles while satisfying the constraints on the interatomic distances of the end residues. Starting from a random conformation, all the dihedral angles are modified at once in each step of the iteration until the distance constraints between the end residues are satisfied. Because of the matrix inversion required, tweak is sometimes numerically unstable, if the matrix loses rank (i.e., has determinant 0 and is therefore uninvertible). Tweak requires that the resulting loop be rotated into place, because the algorithm attempts to satisfy distance constraints between the N- and C-terminal anchor atoms, rather than between the last residue of the moving loop and the fixed anchor. It also does not allow imposing additional constraints on individual residues because modifications to all dihedral angles are computed at once, with strong dependence of each dihedral change on all of the others. It has been used in a number of loop-modeling programs, including Drawbridge (Ring et al. 1992), the Biopolymer program (Tripos, Inc.), and Loopy (Xiang et al. 2002).

Both Modeller (Fiser et al. 2000) and the "scaling relaxation" method of Zheng et al. (1992) build initial conformations that connect the anchors by scaling the size of an initial conformation to fit the anchors, and then gradually returning the loop to normal size through an energy minimization or molecular dynamics procedure. In Modeller, the backbone atoms are built in a straight line from one anchor to the other, whereas in the scaling relaxation method a database loop is used.

Our implementation of the random tweak algorithm and analysis of its limitations led us to examine a variety of so-called inverse kinematics algorithms used in robotics and computer-generated character animation. Forward kinematics methods calculate the positions of object components based on internal and external degrees of freedom, whereas inverse kinematics methods calculate the necessary changes in internal and external degrees of freedom in order for an object component to reach a desired position. Inverse kinematics algorithms are designed to move an "end effector" (e.g., a robotic gripper) to reach for a specific location to pick up an object by changing joint angles and modifying segment lengths. As such, it is essentially the same problem as loop closure in proteins or other molecules, as originally pointed out by Manocha and Zhu (1994; Manocha et al. 1995). Many inverse kinematics algorithms are based on computing the Jacobian and its inverse or pseudoinverse, and hence like tweak are computationally expensive and sometimes numerically unstable (Lander 1998). They rely on changing all joint variables at the same time along a path that will move the end effector toward the target object. In

robotics, the problems of singularities in Jacobian-based methods have been studied extensively (Maciejewski 1990; Merlet 1992). Besides the computational difficulties, one major drawback of Jacobian-based methods is that placing constraints on some degrees of freedom may produce unpredictable results. Capping or zeroing out certain elements of the proposed vector of the changes in degrees of freedom may result in motion of the end effector away from rather than toward the target object.

One algorithm used in robotics that is flexible in allowing constraints to be placed at each step, easy to program, conceptually simple, and computationally fast is "cyclic coordinate descent" (CCD). This algorithm was originally developed as an improved method for solving inverse kinematics problems in robotics (Wang and Chen 1991). CCD is a member of a class of iterative relaxation algorithms known as Jacobi or Gauss-Seidel methods (Briggs et al. 2000). It involves adjusting one degree of freedom at a time to move the end effector toward the target object. This results in one equation in one unknown for each degree of freedom, and hence is analytically very simple and computationally fast. The method is free of singularities and it does not include matrix inversion. It proceeds in iterative fashion along a chain of degrees of freedom, modifying each joint so that the end effector gets as close as possible to the desired position. The equations are able to provide both an optimum setting for the variable and the first and second derivative of the change at the current position so that small increments can be made in preference to large changes, if desired. Given that the optimal change in a parameter in one joint depends only on the current values of the other joint parameters, one can place constraints on any degree of freedom, choosing to restrict their allowed values or place probability distributions on them.

In this paper, we describe the cyclic coordinate descent algorithm, which we have modified for the problem of loop closure in protein structure prediction. In robotics, the end effector is usually a single point and the target position is a single point. In protein modeling, the end effector may be the three backbone atoms of the C-terminal residue of the loop that must be superimposed onto the backbone atoms of the fixed C-terminal anchor of the target model. Therefore, we must also consider the orientation of the end effector as well as its position. In the Materials and Methods section, we describe the algorithm and derive the necessary equations for dihedral angle degrees of freedom and the orientation constraint. There are several possibilities for choice of end effector and target, and we describe one such possible choice and its implementation. In the Results section, we show that CCD can close loops from nearly any starting configuration as long as the chain is long enough to reach from the N-terminal anchor to the C-terminal anchor. We have also explored the use of Ramachandran probability maps as constraints in the CCD closure procedure. This is accomplished by using CCD as a proposal step in a Monte Carlo simulation. The CCD equations provide the move to new values of the backbone dihedrals, and we use Ramachandran map probabilities to determine whether to accept the move. We show that using the Ramachandran constraint does not affect the success rate of loop closure by CCD.

Materials and methods

Our implementation of the CCD method for protein loop closure is an iterative procedure that modifies sequentially each backbone dihedral angle (ϕ and ψ) in each residue that is part of the loop. We define "N-anchor" and "C-anchor" to be the N- and C-terminal anchor residues, respectively, that bracket the loop and remain fixed throughout the calculation. These are illustrated in Figure 1A. The residues are numbered from 0 to n, where the 0-th residue is the N-anchor and the n-th residue is the C-anchor. The calculation begins with some initial configuration of the loop consisting of residues 0 through n. Residue 0 of the loop coincides exactly with the N-anchor, whereas the position of residue n of the loop will not coincide with the position of the fixed C-anchor. The goal is to adjust the backbone dihedrals ϕ and ψ of residues 1 to n so that the backbone atoms N, C_α, and C of the moving residue n are superimposed on the corresponding backbone atoms of the fixed residue n (i.e., the C-anchor).

The initial structure of the loop can be from any source. For instance, it might be constructed from random values for the backbone dihedrals and standard bond lengths and bond angles, or it might be obtained from a database search for loops that approximate the anchors. In either case, we have initial values for all dihedrals, bond lengths, and bond angles for residues 1 through n. In our implementation, starting from residue 0, we build the N atom of residue 1 from the known (or chosen) value of ψ of residue 0, and the C_α atom from ω of residue 1. These atoms remain fixed through the rest of the calculation. The remaining atoms of the loop, beginning with C of residue 1 through C of residue n are built from values of the dihedral angles ϕ,ψ and the initial bond lengths and angles.

Once the initial loop is constructed, the procedure involves changing the values of the backbone dihedrals ϕ and ψ iteratively until the backbone atoms of residue n are superimposed on the fixed backbone atoms of the C-anchor residue. The progress of the loop closing process is assessed by the distances between the backbone atoms of the moving C-terminal residue of the loop and their desired positions in the C-anchor.

As shown in Figure 1B, F_1, F_2, and F_3 are vectors that represent the fixed target positions for the atoms of the C-terminal residue of the loop. The positions of the moving C-terminal residue atoms are represented by M_{01},M_{02},M_{03}, and M_1,M_2,M_3, before and after a change, respectively, in a dihedral angle of any residue in the loop. The rotation axis (containing O_1, O_2, O_3) is given by the direction of the bond corresponding to the dihedral angle that is modified (N—C_α for ϕ, C_α—C for ψ), where O_1, O_2, and O_3 are the footpoints of vectors from the rotation axis to the three atoms of the moving C-terminal anchor.

To accomplish the overlap between the current and desired positions of the atoms, the sum of squared distances, S, should be minimized:

$$S = \left|\overrightarrow{F_1M_1}\right|^2 + \left|\overrightarrow{F_2M_2}\right|^2 + \left|\overrightarrow{F_3M_3}\right|^2 \quad (1)$$

where

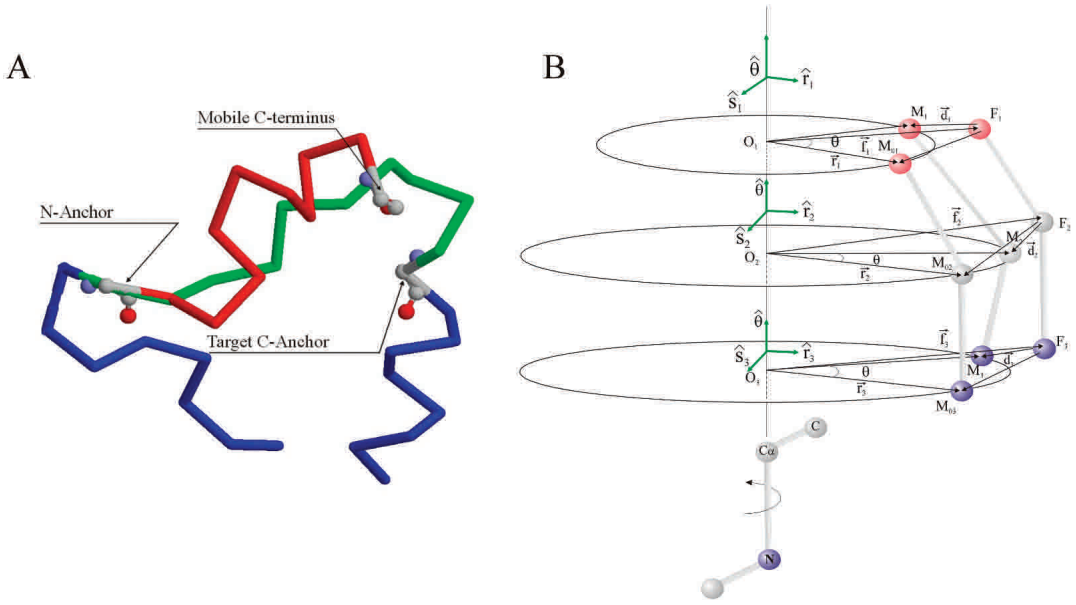

Figure 1. (*A*) C_α trace of a loop before (red) and after (green) closure with the flanking secondary structures (blue). The moving C-terminal anchor and the fixed C-terminal anchor are indicated. The loop closure problem is to adjust the dihedral angle degrees of freedom of the loop so that the moving C-terminal anchor is superimposed on the fixed C-terminal anchor. (*B*) Schematic of the CCD algorithm. Variables are defined in the text.

$$\vec{F}_1M_1 = \vec{O}_1M_1 - \vec{O}_1F_1 \tag{2}$$

Notating $\vec{O}_1M_{01} = \vec{r}_1$ and $\vec{O}_1F_1 = \vec{f}_1$ and defining a local orthogonal system of coordinates $\hat{r}_1$, $\hat{s}_1$, $\hat{\theta}$, where $\hat{r}_1$ is the unit vector of $\vec{r}_1$ and $\hat{\theta}$ is the unit vector along the rotation axis, we can write:

$$\vec{O}_1M_1 = r_1 \cos\theta\hat{r}_1 + r_1 \sin\theta\hat{s}_1 \tag{3}$$

From equations 2 and 3, it follows that

$$\vec{F}_1M_1 = r_1 \cos\theta\hat{r}_1 + r_1 \sin\theta\hat{s}_1 - \vec{f}_1 \equiv \vec{d}_1 \tag{4}$$

with similar equations for the second and third atoms.

Calculating the squared distances between the moving atoms and the fixed target atoms, we obtain:

$$\begin{aligned}
\left|\vec{d}_1\right|^2 &= r_1^2 + f_1^2 - 2r_1\cos\theta\,(\vec{f}_1\cdot\hat{r}_1) - 2r_1\sin\theta\,(\vec{f}_1\cdot\hat{s}_1)\\
\left|\vec{d}_2\right|^2 &= r_2^2 + f_2^2 - 2r_2\cos\theta\,(\vec{f}_2\cdot\hat{r}_2) - 2r_2\sin\theta\,(\vec{f}_2\cdot\hat{s}_2)\\
\left|\vec{d}_3\right|^2 &= r_3^2 + f_3^2 - 2r_3\cos\theta\,(\vec{f}_3\cdot\hat{r}_3) - 2r_3\sin\theta\,(\vec{f}_3\cdot\hat{s}_3)
\end{aligned} \tag{5}$$

The first-order derivative for S is

$$\frac{dS}{d\theta} = \frac{d\left(\left|\vec{d}_1\right|^2\right)}{d\theta} + \frac{d\left(\left|\vec{d}_2\right|^2\right)}{d\theta} + \frac{d\left(\left|\vec{d}_3\right|^2\right)}{d\theta} \tag{6}$$

where for $i = 1, 2, 3$ we have

$$\frac{d\left(\left|\vec{d}_i\right|^2\right)}{d\theta} = 2r_i\sin\theta\,(\vec{f}_i\cdot\hat{r}_i) - 2r_i\cos\theta\,(\vec{f}_i\cdot\hat{s}_i) \tag{7}$$

Setting $dS/d\theta = 0$, we obtain $\tan\alpha$, where α is the rotation angle that will yield an extreme value for the sum of square distances described above.

$$\tan\alpha = \frac{(\vec{f}_1\cdot\hat{s}_1)r_1 + (\vec{f}_2\cdot\hat{s}_2)r_2 + (\vec{f}_3\cdot\hat{s}_3)r_3}{(\vec{f}_1\cdot\hat{r}_1)r_1 + (\vec{f}_2\cdot\hat{r}_2)r_2 + (\vec{f}_3\cdot\hat{r}_3)r_3} \tag{8}$$

Inverting the tangent will produce two values for α that are π radians apart. The correct one is that which produces a positive value of the second derivative of S, which is easily derived from equations 6 and 7.

In practice we obtain α in a different way. With S of the form

$$S = a - b\cos\theta - c\sin\theta, \tag{9}$$

multiplying the last two terms by

$$\sqrt{b^2+c^2}\,/\,\sqrt{b^2+c^2},$$

and defining

$$\cos\alpha = \frac{b}{\sqrt{b^2+c^2}},$$

$$\sin\alpha = \frac{c}{\sqrt{b^2+c^2}} \tag{10}$$

S can be written as

$$S = a - \sqrt{b^2 + c^2}\cos(\theta - \alpha) \quad (11)$$

When $\theta = \alpha$, S is a minimum. This is the same solution as equation 8, except that we now have sine and cosine explicitly defined. We use the atan2(y, x) function of the C programming language to return θ in the correct quadrant, rather than making use of the second-order derivative test.

Test set

To test our algorithm, we selected a set of 2752 loops from 366 X-ray crystallographic structures in the PDB. The selected loops belong to structures that have been solved to a resolution better than 1.6 Å and have mutual sequence identity <20%. The list of structures was obtained from the PISCES server (formerly the CulledPDB server, now available at http://www.fccc.edu/research/labs/dunbrack/pisces). The chosen loops were identified as having coil structure by the Stride program (Frishman and Argos 1995). None of the loops in the test set are adjacent to disordered residues as determined by our validation program S2C (http://www.fccc.edu/research/labs/dunbrack/s2c).

Ramachandran probability maps

We derived Ramachandran probabilities from data used to build our backbone-dependent rotamer libraries (May 2002 release; Dunbrack Jr. and Cohen 1997). Pairs of ϕ,ψ dihedral angles were weighted with a Gaussian function, and counts were produced in 10° bins in ϕ and ψ over the entire Ramachandran map.

Implementation

We have implemented CCD in object-oriented C++ under RedHat Linux 7.3. All calculations were performed on an AMD1800+ MP processor.

Results

For each loop in our test set, we generated 100 random loops by drawing values for the ϕ and ψ dihedral angles randomly from PDB structures used to build our backbone-dependent rotamer library (Dunbrack Jr. and Cohen 1997). We used random ϕ,ψ from the PDB rather than random values from the interval 0°–360° to produce more protein-like starting conformations. Loops were built starting from the N-anchor residue Cartesian coordinates from the structure as described in Materials and Methods. The last residue of the random loop structure was built with the crystallographic bond lengths and bond angles, so that the RMS of the superposition of the moving and fixed C-anchors would not depend on differences in these parameters. We consider the loop closed when the RMS of the N, C_α, and C atoms of the moving and fixed C-anchor residues is <0.08 Å. The maximum number of CCD iterative cycles was limited to 5000.

The first implementation of CCD entailed using equation 10 to calculate the change in each dihedral along the chain and accepting the proposed move with probability 1. This algorithm is denoted "CCD No Constraint" in Table 1. Of the 275,200 loops closed (100 random conformations of each of 2752 loops in the test set), 99.79% of them closed to within 0.08 Å RMS in fewer than 5000 steps, where each step consists of a single cycle through all dihedral angles of the loop. Most loops closed in fewer than 200 steps (see below).

To examine the effect of adding a constraint to CCD, we used Ramachandran probability maps to determine whether moves proposed by equation 10 would be accepted in each step. The algorithm proposed a change in ϕ and, based on the new ϕ, proposed a new value for ψ. The new ϕ,ψ position was accepted with probability 1.0 if the probability of the new ϕ,ψ was higher than the current value. It was accepted with probability p_{new}/p_{old} if the new probability was lower than the current value. The results over the same set of 275,200 random conformations are also shown in Table 1 (labeled "CCD Ramachandran Map"), and demonstrate that the Ramachandran constraint has essentially no effect on the loop closure rate.

We investigated situations in which the generated loops did not close within 5000 steps. For each unclosed loop, the RMS of N, C_α, and C atoms of the C-terminus residue in the simulated loop and C-anchor was calculated. The distributions of the number of loops as a function of the RMS are shown in Figure 2. The figure shows that the greatest majority of the loops that were not able to close within the imposed error margin (0.08 Å) are within 0.1 Å RMS from the target position, in both versions of the algorithm. There is also no significant difference between the Ramachandran probability version of the algorithm and the unconstrained one, with respect to the RMS distributions.

The lowest closure rates were for very short loops of 4 or 5 amino acids. We examined loops that failed to close, and in every case these were extended conformations. CCD con-

Table 1. *Results of loop closure trials*

	CCD no constraint			CCD Ramachandran map		
Loop length	No. of loops	No. of unclosed loops	Percentage of closed loops	No. of loops	No. of unclosed loops	Percentage of closed loops
4	58,500	886	98.49	58,500	848	98.55
5	54,400	208	99.62	54,400	213	99.61
6	37,400	80	99.79	37,400	90	99.76
7	29,800	18	99.94	29,800	15	99.95
8	24,500	13	99.95	24,500	16	99.93
9	21,000	4	99.98	21,000	2	99.99
10	14,400	3	99.98	14,400	2	99.99
11	11,900	6	99.95	11,900	2	99.98
12	9800	2	99.98	9800	0	100
13	7500	0	100	7500	0	100
14	6000	0	100	6000	0	100
Total	275,200	1220	99.79	275,200	1188	99.80

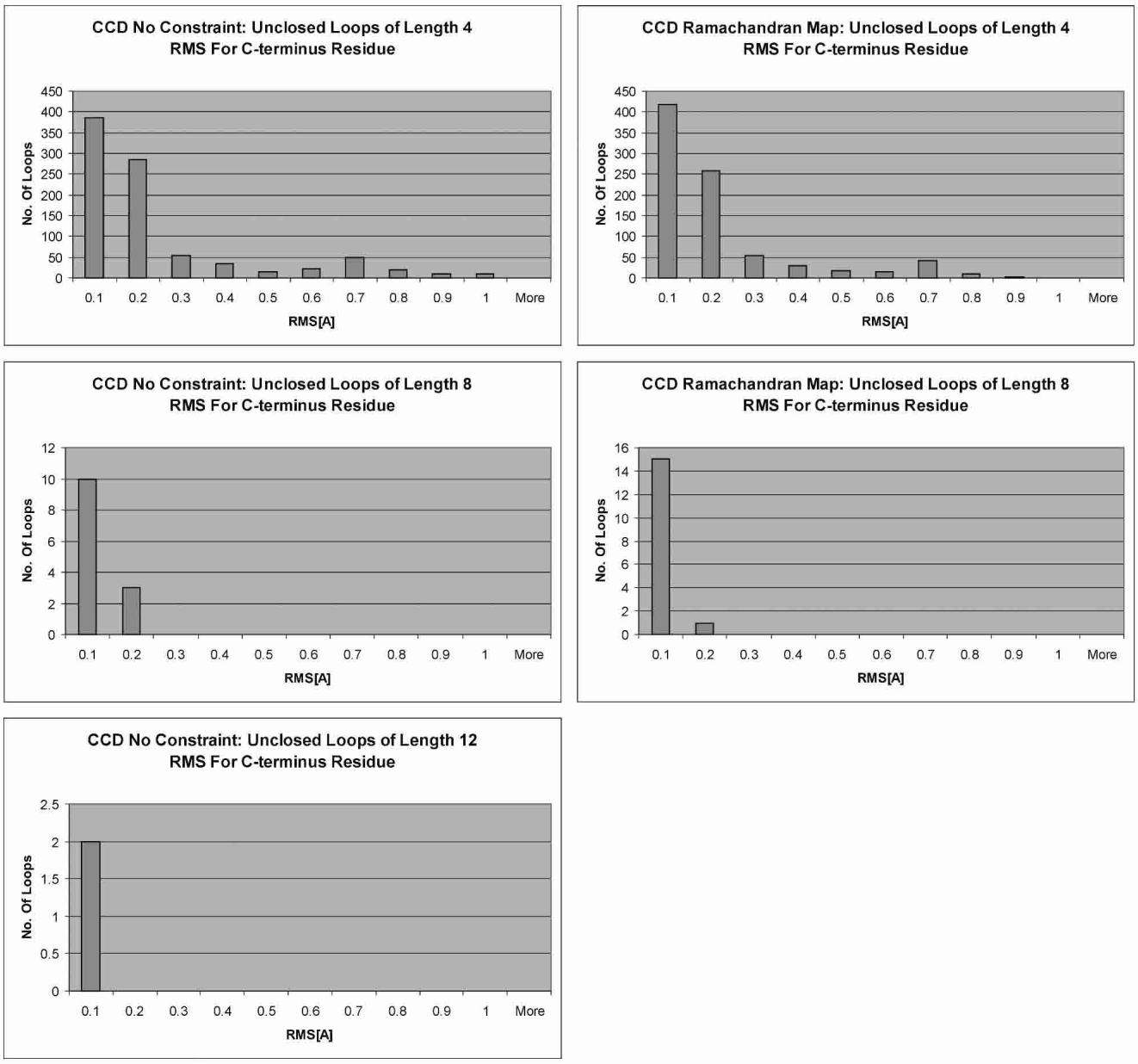

Figure 2. Histograms of RMS values for unclosed loops at loop lengths of 4, 8, and 12 for the CCD No Constraint and the CCD Ramachandran Map algorithms. Note that all loops of length 12 closed for the CCD Ramachandran Map algorithm, so there is no plot.

verged in a local minimum for every dihedral of the loop. A simple Monte Carlo step that proposes moves that increase the RMS can be added in these situations to move the algorithm out of the local minimum. It should also be noted that for these extended loops, most trials actually closed, and only a small number failed to converge. Histograms for the number of steps required to close the loops in the input data set are shown in Figure 3. The large majority of trials close within 100–500 steps. For all loop lengths, the number of steps required with the Ramachandran map constraint is higher than without the constraint, as expected.

CCD has been designed to be used with any loop prediction algorithm that generates reasonable trial structures, coupled with an energy function to identify the best loops. As such, it is not a prediction method on its own, nor is it a sampling algorithm per se but, rather, a component of one. Nevertheless, we were interested to determine how many random loop structures would need to be built and closed with the generation procedure described in Materials and Methods to obtain a reasonable RMS to the real structure. For this purpose, we chose 10 loops at lengths 4, 8, and 12, for a total of 30 loops. For each of the loops we generated

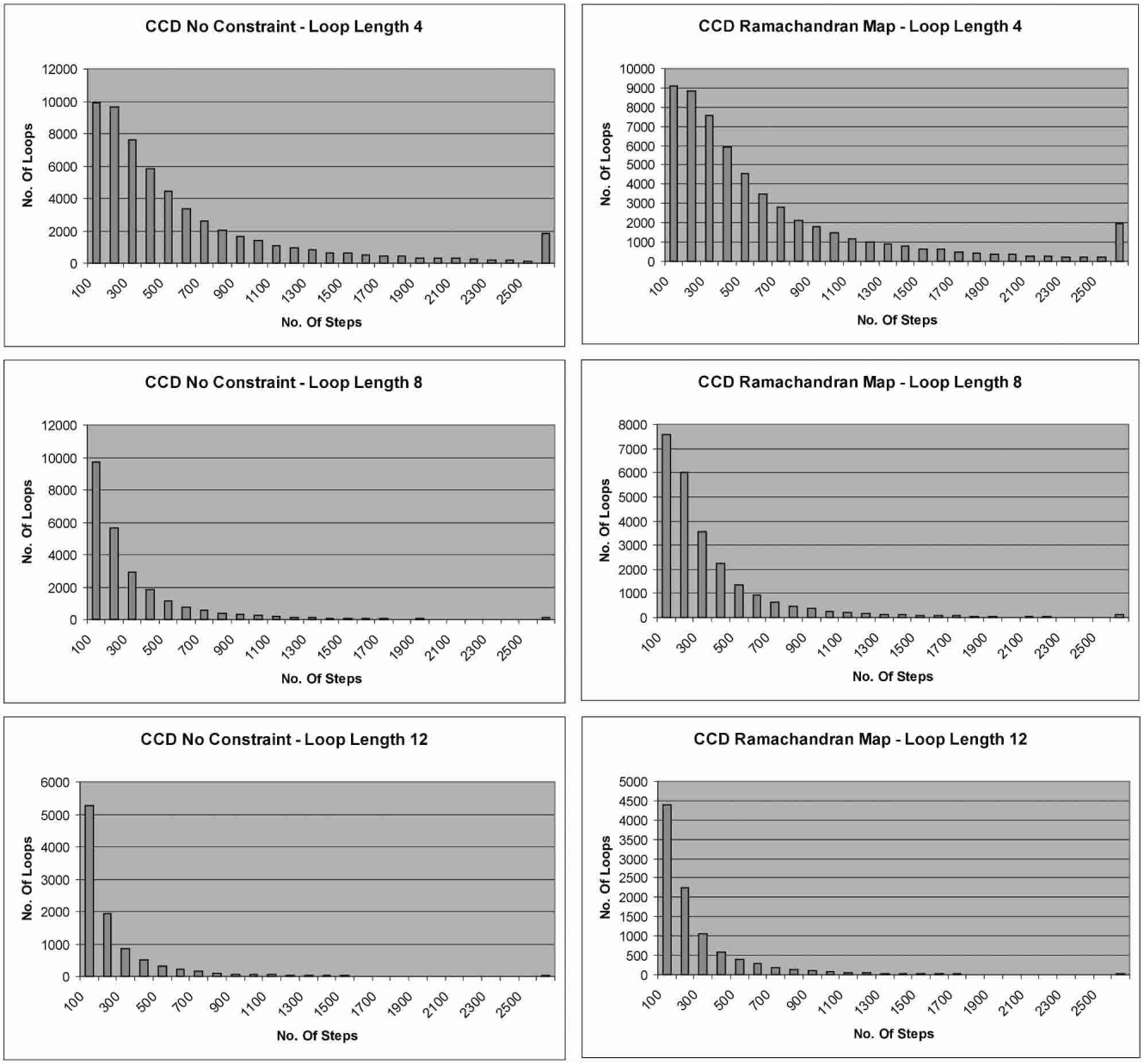

Figure 3. Histograms of the number of steps for loop closure at loop lengths 4, 8, and 12 for the CCD No Constraint and the CCD Ramachandran Map algorithms.

5000 random loops and closed them, using the CCD Ramachandran Map algorithm. In Table 2, we show the minimum RMS achieved for these 30 loops. The average for loops of 4, 8, and 12 amino acids is 0.56, 1.59, and 3.04 Å, respectively. Examples of these minimum RMS conformations are shown in Figure 4. Although the RMS is low in Figure 4C for the 12-amino-acid loop, other samples might better reflect the positions of backbone carbonyls and NH groups relative to the whole protein structure.

We investigated the CCD Ramachandran map constraint method to determine whether loops closed from the same starting conformation would converge to the same structure, cluster into groups, or be distributed randomly. The sequence of random numbers used to determine whether moves are accepted or not was seeded differently in each run, resulting in different closed structures from the same initial structure. We used a loop from PDB entry 1egu (Li et al. 2000), residues 508–519 of length 12, closed it 500 times from the same conformation, and calculated the RMS between each pair in this set. It is useful to compare the distributions of RMS values among this set, with the RMS values among a set of 500 closures of the same loop, but starting from different initial conformations for each trial. The comparison is shown in Figure 5. Loops starting

Table 2. *Minimum RMS from X-ray structure in 5000 trials per loop of CCD Ramachandran map algorithm*

Length 4		Length 8		Length 12	
Loop	Min RMS	Loop	Min RMS	Loop	Min RMS
1dvjA_20-23	0.606	1cruA_85-92	1.753	1cruA_358-369	2.538
1dysA_47-50	0.676	1ctqA_144-151	1.344	1ctqA_26-37	2.487
1eguA_404-407	0.675	1d8wA_334-341	1.506	1d4oA_88-99	2.328
1ej0A_74-77	0.337	1ds1A_20-27	1.581	1d8wA_46-57	4.827
1i0hA_123-126	0.616	1gk8A_122-129	1.684	1ds1A_282-293	3.042
1id0A_405-408	0.671	1i0hA_145-152	1.351	1dysA_291-302	2.478
1qnrA_195-198	0.491	1ixh_106-113	1.605	1eguA_508-519	2.137
1qopA_44-47	0.627	1lam_420-427	1.604	1f74A_11-22	2.715
1tca_95-98	0.393	1qopB_14-21	1.849	1q1wA_31-42	3.378
1thfD_121-124	0.495	3chbD_51-58	1.659	1qopA_178-189	4.568
Avg. min RMS	0.559	Avg. min RMS	1.594	Avg. min RMS	3.050

from the same conformation do cluster, as demonstrated by the peak at RMS near 0 Å. Loops starting from different conformations do not converge to the same structure. The RMS values approximately follow a gamma distribution.

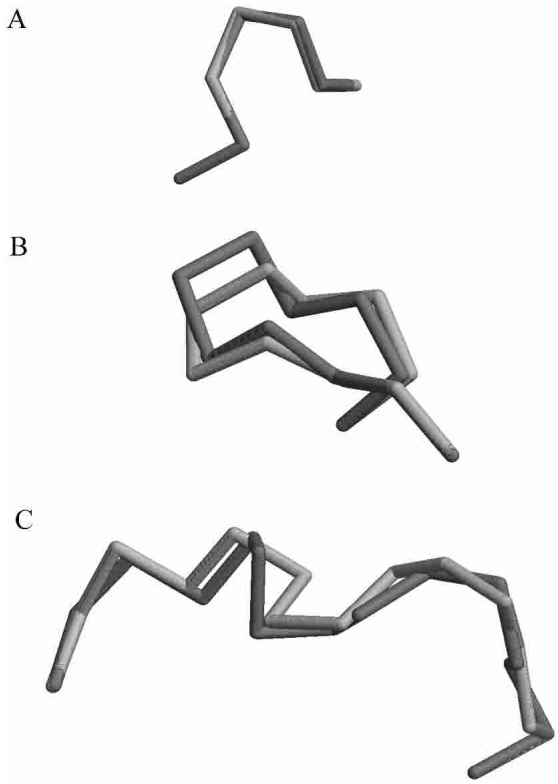

Figure 4. C_α renderings of the lowest RMS loop generated from 5000 trials of the CCD Ramachandran Map method for loops of 4, 8, and 12 amino acids, compared with the X-ray structures (dark figures). (*A*) Loop 1ej0A_74–77, (*B*) loop 1ctqA_144–151, (*C*) loop 1eguA_508–519.

We compared CCD to the random tweak method (Shenkin et al. 1987) used in some loop prediction algorithms (Xiang et al. 2002). In our version of random tweak, we set constraints on the distances of 0 Å between the moving C-terminal anchor N, C_α, and C atoms and the corresponding atoms of the fixed C-anchor. Note that this is different from the original description, which fixes distances between the N- and C-terminal C_α atoms. We ran the same test set shown in Table 2 for 8 residue loops, consisting of 500 initial configurations of 10 loops, for a total of 5000 trials for both tweak and CCD No Constraint. We used the same final RMS criterion for a closed loop of 0.08 Å and 5000 rounds of each algorithm maximum. CCD was able to close all 5000 of these loops in ~7 min, whereas random tweak closed 4841 loops in 40 min, failing on 159 loops.

Finally, we further investigated the computation time needed by the CCD algorithm itself, without the time needed to read the initial conformation and the full Ramachandran probability maps for the 20 amino acids from the disk for each closure. We used the same set of 10 loops of length 4, 8, and 12 listed in Table 2, but this time generated 500 random conformations within the program, thus reading in only a single initial conformation and reading in the probability map only once. We generated 500 random conformations in this manner and closed them, with two different RMS cutoffs. With an error margin of 0.08 Å, the average computing time was 0.031, 0.037, and 0.023 sec per loop for loops of lengths 4, 8, and 12, respectively. With a looser cutoff of 0.16 Å, the times would be significantly lower. For instance, for 8 amino acid loops, the average computing time was 0.026 sec per loop. It is interesting to note that CCD takes less time for longer loops, because these loops have more degrees of freedom and more solutions to the loop closure problem.

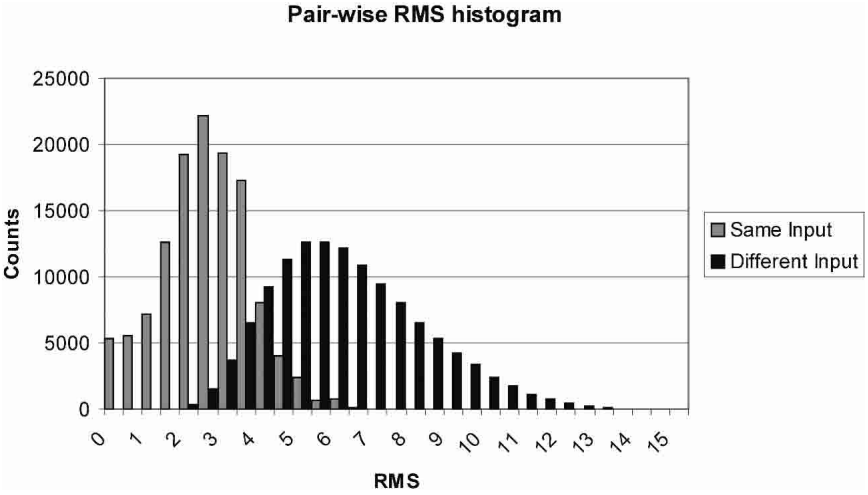

Figure 5. Comparison of distribution of RMS among conformations generated from the same initial structure (light bars) and different initial structures (dark bars). PDB entry 1egu, residues 508–519, was used as the test case.

Discussion

Loop closure is a component of a number of protein structure prediction problems, including various approaches to homology modeling as well as Monte Carlo simulations of protein folding. Because most structure prediction methods proceed by producing large numbers of trial configurations, a fast loop closure method is of significant importance. Although several algorithms have been presented previously, each suffers from a number of drawbacks. These include lack of convergence, numerical instability, and cumbersome implementations. We have described a very simple method for loop closure borrowed from robotics that is easy to understand and implement.

As mentioned in the Results section, CCD is not meant to be a sampling strategy on its own but, rather, is to be used with any method that generates unclosed trial conformations. In this paper, we have generated trial conformations by drawing random values for ϕ,ψ from X-ray data from the PDB. These data were taken from loop residues and were specific for each amino acid type. The procedure as described here does not have any steric bump checks or internal energy evaluations of any kind, other than the Ramachandran probabilities. Nevertheless, this procedure was able to generate loops on average minimum RMS values of 0.56, 1.59, and 3.04 Å from the native structure for loops of 4, 8, and 12 residues. This compares favorably with some other sampling strategies, such as that of Tosatto et al. (2002), who used a divide-and-conquer method to generate average minimum RMS values of 1.0, 2.22, and 3.5 Å for loops of lengths 4, 8, and 12. Sudarsanam et al. (1995) used a database of ψ_i,ϕ_{i+1} pairs to model loops and achieved lowest RMSD values of 1.2–1.3 Å in 10,000 simulations for an 8-amino-acid loop. However, these loops were not closed. By including a CHARMM-based energy function during loop construction and simulation, Fiser et al. (2000) were able to achieve lowest RMSDs (regardless of energy) of 0.70, 0.93, and 1.93 for three 8-amino-acid loops. It remains to be seen whether in designing a loop prediction strategy incorporating CCD it is better to include energy evaluations during the CCD closure procedure, or to build large numbers of samples and evaluate their energy in the context of the protein afterward. CCD can be easily modified to include other constraints such as avoidance of collisions. Techniques from robot motion planning would probably be most suitable for this purpose (Singh et al. 1999).

CCD's advantages compared with Jacobian-based methods are its simplicity, ease of implementation, speed, and lack of singularities (Welman 1993). One disadvantage in our present implementation is that the algorithm favors large changes in the first residues of the loop. If the loop can be nearly closed with manipulations of the first few residues, then the other residues will barely move at all. This probably occurs fairly rarely. In any case, to preserve similarity to the initial configuration and to even out the changes in the dihedrals across the whole loop, one can impose limits on the change in dihedral angles at each step. Because we have an expression for the distance (and its derivative) of the moving C-anchor to the fixed C-anchor, we can choose to make small moves toward an RMS of 0, rather than propose the full CCD step to the minimum value of S in equation 11. Another disadvantage is that CCD may occasionally get stuck in a local minimum, when solving equation 11 for each dihedral results in no change in configuration. The method can be modified to check for this and to add a step that will move the moving C-anchor residue away from the target.

CCD is a good example of crossover of algorithms from one field to another that is a hallmark of bioinformatics and computational biology. The inverse kinematics problem is a staple of robotics, and the cyclic coordinate descent algorithm described here is one of several methods that are likely to be borrowed from this field in structural biology. Other recent examples include the use of robot motion planning and probabilistic roadmaps in protein folding (Singh et al. 1999; Apaydin et al. 2002; Brutlag et al. 2002), analytical inverse kinematics approaches applied to protein loop closures of six dihedral degrees of freedom or fewer (Manocha and Zhu 1994; Manocha et al. 1995), and a randomized kinematics search for loop closure in the drug design problem (Lavalle et al. 2000). It is likely that there will be more interactions between these disciplines in the future.

Acknowledgments

We gratefully acknowledge support from NIH Grants R01-HG2302 to R.L.D. and CA06972 to Fox Chase Cancer Center. A.A.C. is an NIH Postdoctoral Trainee.

The publication costs of this article were defrayed in part by payment of page charges. This article must therefore be hereby marked "advertisement" in accordance with 18 USC section 1734 solely to indicate this fact.

References

Apaydin, M.S., Guestrin, C.E., Varma, C., Brutlag, D.L., and Latombe, J.C. 2002. Stochastic roadmap simulation for the study of ligand–protein interactions. *Bioinformatics* **18 Suppl.:** S18–S26.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. 2000. The Protein Data Bank. *Nucleic Acids Res.* **28:** 235–242.

Bower, M.J., Cohen, F.E., and Dunbrack Jr., R.L. 1997. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: A new homology modeling tool. *J. Mol. Biol.* **267:** 1268–1282.

Briggs, W.L., Henson, V.E., and McCormick, S.F. 2000. *A multigrid tutorial.* SIAM, Philadelphia.

Bruccoleri, R.E. and Karplus, M. 1987. Prediction of the folding of short polypeptide segments by uniform conformational sampling. *Biopolymers* **26:** 137–168.

Brutlag, D., Apaydin, S., Guestrin, C., Hsu, D., Varma, C., Singh, A., and Latombe, J.C. 2002. Using robotics to fold proteins and dock ligands. *Bioinformatics* **18 Suppl.:** S74.

Dunbrack Jr., R.L. 1999. Comparative modeling of CASP3 targets using PSI-BLAST and SCWRL. *Proteins* **Suppl. 3:** 81–87.

Dunbrack Jr., R.L. and Cohen, F.E. 1997. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci.* **6:** 1661–1681.

Fine, R.M., Wang, H., Shenkin, P.S., Yarmush, D.L., and Levinthal, C. 1986. Predicting antibody hypervariable loop conformations. II: Minimization and molecular dynamics studies of MCPC603 from many randomly generated loop conformations. *Proteins* **1:** 342–362.

Fiser, A., Do, R.K., and Sali, A. 2000. Modeling of loops in protein structures. *Protein Sci.* **9:** 1753–1773.

Frishman, D. and Argos, P. 1995. Knowledge-based protein secondary structure assignment. *Proteins* **23:** 566–579.

Jones, D.T., Tress, M., Bryson, K., and Hadley, C. 1999. Successful recognition of protein folds using threading methods biased by sequence similarity and predicted secondary structure. *Proteins* **Suppl3:** 104–111.

Karplus, K., Barrett, C., and Hughey, R. 1998. Hidden Markov models for detecting remote protein homologies. *Bioinformatics* **14:** 846–856.

Lander, J. 1998. Making kine more flexible. *Game Developer* **1 (Nov.):** 15–22.

Lavalle, S.M., Finn, P.W., Kavraki, L.E., and Latombe, J.-C. 2000. A randomized kinematics-based approach to pharmacophore-constrained conformational search and database screening. *J. Comp. Chem.* **21:** 731–747.

Li, S., Kelly, S.J., Lamani, E., Ferraroni, M., and Jedrzejas, M.J. 2000. Structural basis of hyaluronan degradation by *Streptococcus pneumoniae* hyaluronate lyase. *EMBO J.* **19:** 1228–1240.

Liang, S. and Grishin, N.V. 2002. Side-chain modeling with an optimized scoring function. *Protein Sci.* **11:** 322–331.

Maciejewski, A.A. 1990. Dealing with ill-conditioned equations of motion for articulated figures. *IEEE Comp. Graph. App.* **10:** 233–242.

Manocha, D. and Zhu, Y. 1994. Kinematic manipulation of molecular chains subject to rigid constraints. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **2:** 285–293.

Manocha, D., Zhu, Y., and Wright, W. 1995. Conformational analysis of molecular chains using nano-kinematics. *Comput. Appl. Biosci.* **11:** 71–86.

Mendes, J., Nagarajaram, H.A., Soares, C.M., Blundell, T.L., and Carrondo, M.A. 2001. Incorporating knowledge-based biases into an energy-based side-chain modeling method: Application to comparative modeling of protein structure. *Biopolymers* **59:** 72–86.

Merlet, J.-P. 1992. Geometry and kinematic singularities of closed-loop manipulators. *J. Lab. Robotics Automation* **4:** 85–96.

Moult, J. and James, M.N.G. 1986. An algorithm for determining the conformation of polypeptide segments in proteins by systematic search. *Proteins* **1:** 146–163.

Ring, C.S., Kneller, D.G., Langridge, R., and Cohen, F.E. 1992. Taxonomy and conformational analysis of loops in proteins. *J. Mol. Biol.* **224:** 685–699; Erratum **227:** 977.

Sali, A. and Blundell, T.L. 1993. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234:** 779–815.

Sauder, J.M. and Dunbrack Jr., R.L. 2000. Beyond genomic fold assignment: Rational modeling of proteins of biological interest. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8:** 296–306.

Sauder, J.M., Arthur, J.W., and Dunbrack Jr., R.L. 2000. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins* **40:** 6–22.

Shenkin, P.S., Yarmush, D.L., Fine, R.M., Wang, H.J., and Levinthal, C. 1987. Predicting antibody hypervariable loop conformation. I. Ensembles of random conformations for ringlike structures. *Biopolymers* **26:** 2053–2085.

Simons, K.T., Bonneau, R., Ruczinski, I., and Baker, D. 1999. Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins* **37:** 171–176.

Singh, A.P., Latombe, J.C., and Brutlag, D.L. 1999. A motion planning approach to flexible ligand binding. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **7:** 252–261.

Sudarsanam, S., DuBose, R.F., March, C.J., and Srinivasan, S. 1995. Modeling protein loops using a ϕ_{i+1},ψ_i dimer database. *Protein Sci.* **4:** 1412–1420.

Tosatto, S.C., Bindewald, E., Hesser, J., and Manner, R. 2002. A divide and conquer approach to fast loop modeling. *Protein Eng.* **15:** 279–286.

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

Wang, L.T. and Chen, C.C. 1991. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Trans. Robotics Automation* **7:** 489–499.

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

Welman, C. 1993. Inverse kinematics and geometric constraints for articulated figure manipulation. In *School of computing science*, pp. 77. Simon Fraser University, Burnaby, BC, Canada.

Wheeler, D.L., Church, D.M., Lash, A.E., Leipe, D.D., Madden, T.L., Pontius, J.U., Schuler, G.D., Schriml, L.M., Tatusova, T.A., Wagner, L., et al. 2002. Database resources of the National Center for Biotechnology Information: 2002 update. *Nucleic Acids Res.* **30:** 13–16.

Xiang, Z. and Honig, B. 2001. Extending the accuracy limits of prediction for side-chain conformations. *J. Mol. Biol.* **311:** 421–430.

Xiang, Z., Soto, C.S., and Honig, B. 2002. Evaluating conformational free energies: The colony energy and its application to the problem of protein loop prediction. *Proc. Natl. Acad. Sci.* **99:** 7432–7437.

Zheng, Q., Rosenfeld, R., Vajda, S., and DeLisi, C. 1992. Loop closure via bond scaling and relaxation. *J. Comp. Chem.* **14:** 556–565.