Protein Engineering vol.15 no.4 pp.279–286, 2002

A divide and conquer approach to fast loop modeling

Silvio C.E.Tosatto[1,2], Eckart Bindewald, Jürgen Hesser and Reinhard Männer

Institute for Computational Medicine and Chair for Computer Science V, Universität Mannheim, B 6, 26, 68131 Mannheim, Germany

[2]Present address: CRIBI Biotechnology Centre, University of Padova, V. le G. Colombo 3, I-35121 Padova, Italy

[1]To whom correspondence should be addressed.
E-mail: silvio@cribi.unipd.it

We describe a fast *ab initio* method for modeling local segments in protein structures. The algorithm is based on a divide and conquer approach and uses a database of precalculated look-up tables, which represent a large set of possible conformations for loop segments of variable length. The target loop is recursively decomposed until the resulting conformations are small enough to be compiled analytically. The algorithm, which is not restricted to any specific loop length, generates a ranked set of loop conformations in 20–180 s on a desktop PC. The prediction quality is evaluated in terms of global RMSD. Depending on loop length the top prediction varies between 1.06 Å RMSD for three-residue loops and 3.72 Å RMSD for eight-residue loops. Due to its speed the method may also be useful to generate alternative starting conformations for complex simulations.

Keywords: homology modeling/polypeptide conformations/protein folding/structure prediction

Introduction

Determination of protein structures that have not been solved experimentally is frequently done by comparative modeling techniques (Moult *et al.*, 1999). Copying parts of the target structure, which are assumed to be superimposable, from a known protein structure serves as a framework. Structurally variable regions, referred to as loops, have to be treated separately. Because loops often show the greatest variation in amino acid sequence and are usually less restrained in conformation than the core regions, they cannot easily be taken from the parent structure. Their prediction remains one of the main problems in comparative protein modeling (Moult *et al.*, 1999). One problem is the generation of a good set of alternative structures for evaluation with a scoring or energy function. A number of different approaches have been investigated in the literature to tackle this problem, which can be divided in at least three categories: analytical, optimization and database methods.

Analytical methods to predict the conformation of short peptides date back to the pioneering work of Go and Scheraga (Go and Scheraga, 1970). It is possible to predict the conformation of fragments with up to six rotable torsion angles using rigid geometry, i.e. keeping idealized bond lengths and bond angles, by solving a set of equations representing geometric transformations. A number of publications have further addressed this problem (Bruccoleri and Karplus, 1985; Palmer and Scheraga, 1991; Manocha *et al.*, 1995; Wedemeyer and Scheraga, 1999), but the results show that no generalized analytical solution beyond six torsion angles is possible (Go and Scheraga, 1970; Palmer and Scheraga, 1991). Bruccoleri and Karplus (Bruccoleri and Karplus, 1987) have extended the approach, solving small fragments analytically and enumerating the solutions of larger ones. Combinatorial approaches have been studied by several groups (Moult and James, 1986; Bruccoleri *et al.*, 1988; Brower *et al.*, 1993; Bruccoleri, 1993; Fidelis *et al.*, 1994; Pedersen and Moult, 1995; Deane and Blundell, 2000). A discretization of solution space is required to limit the combinatorial explosion. A restricted set of (ϕ,ψ) torsion angles is used to approximate all possible conformations. This ranges from uniform conformational sampling to distributions biased towards more populated regions of the (ϕ,ψ) map. In addition, techniques to limit the combinatorial explosion have been used, e.g. pruning parts of the search tree which are too far apart to be spanned. The search algorithm can either generate the conformations on the fly or separately from modeling. Deane and Blundell (Deane and Blundell, 2000) have presented an interesting fast *ab initio* method to predict loop conformations up to eight residues in length. They use a set of eight carefully chosen (ϕ,ψ) torsion angles, representing over 96% of all possible five-residue fragments with <1 Å RMSD, to generate a database enumerating all combinations up to 12 residues in length. A two-residue overlap on each side of the loop is used to select fitting fragments, allowing up to eight residues to be predicted. The average RMSD ranges between 1.3 Å for three-residue loops and 3.9 Å for eight-residue loops.

Various methods relying on local optimization such as the minimum perturbation random tweak (Fine *et al.*, 1986; Shenkin *et al.*, 1987; Smith and Honig, 1994), local moves (Eloffsson *et al.*, 1995), importance sampling by local minimization of randomly generated conformations (Lambert and Scheraga, 1989a,b,c) and global energy minimization by mapping a trajectory of local minima (Dudek and Scheraga, 1990; Dudek *et al.*, 1998) were also used. Other methods relate to the optimization of an energy function. These include molecular dynamics simulations (Bruccoleri and Karplus, 1990; Tanner *et al.*, 1992; Rao and Teeter, 1993; Nakajima *et al.*, 2000), Monte Carlo and molecular dynamics (Rapp and Friesner, 1999), biased probability Monte Carlo search (Abagyan and Totrov, 1994; Evans *et al.*, 1995; Thanki *et al.*, 1997), Monte Carlo with simulated annealing (Higo *et al.*, 1992; Carlacci and Englander, 1993, 1996; Collura *et al.*, 1993; Vasmatzis *et al.*, 1994; Fiser *et al.*, 2000), scaling relaxation and multiple copy sampling (Rosenfeld *et al.*, 1993; Zheng *et al.*, 1993a,b, 1994; Rosenbach and Rosenfeld, 1995; Zheng and Kyle, 1994, 1996) and self-consistent mean field optimization (Koehl and Delarue, 1995). The resulting loop conformations may cover only a subset of the solution space and are not necessarily close to the native structure.

With the increasing number of structures deposited in the Protein Data Bank (PDB) (Abola *et al.*, 1997) it is possible to model unknown loop conformations according to loop structures of known proteins. This was first developed by Jones and Thirup (Jones and Thirup, 1986), who selected fragments from the PDB for electron density fitting. Similar approaches have been used in comparative modeling (Chothia and Lesk, 1987; Claessens *et al.*, 1989; Summers and Karplus, 1990; Tramontano and Lesk, 1992; Levitt, 1992; Topham *et al.*, 1993; Lessel and Schomburg, 1994; Martin and Thornton, 1996; Li *et al.*, 1999). Fragments are selected from a database of many known structures based on overlap with the framework on both ends and sorted according to geometric criteria or sequence similarity. Databases usually contain longer polypeptide fragments to increase predictive power. Sudarsanam *et al.* (Sudarsanam *et al.*, 1995) use a database of all possible dimers to construct loops in a similar way to enumerative methods. The overlap between fragment and framework alone is unlikely to yield satisfactory results (Tramontano and Lesk, 1992). Van Vlijmen and Karplus (Van Vlijmen and Karplus, 1997) have shown that the results of database methods can be improved by subsequent optimization and ranking using the CHARMM energy function (MacKerell *et al.*, 1998). An algorithm combining database searches with *ab initio* methods has been proposed by Martin *et al.* (Martin *et al.*, 1989). They use a database of backbone conformations to predict long loops. For short loops, and the central part of loops predicted from the database, they rely on the *ab initio* method of Bruccoleri and Karplus (Bruccoleri and Karplus, 1985).

Database methods are able to approximate most of the antibody hypervariable loops quite closely, suggesting that these proteins form a specific sub-space of foldings based on certain ‘key residues’ (Chothia and Lesk, 1986, 1987). This concept has been generalized to determine the conformation of the loop, with limited success (Ring *et al.*, 1992; Oliva *et al.*, 1997; Rufino *et al.*, 1997; Wojcik *et al.*, 1999). Antibody loops have been found to form similar structures, allowing strict classification (Chothia and Lesk, 1987; Chothia *et al.*, 1989; Martin and Thornton, 1996; Morea *et al.*, 1998). Many groups have developed classification methods for loops (Sibanda and Thornton, 1985; Kwasigroch *et al.*, 1996; Sun and Jiang, 1996; Geetha and Munson, 1997; Oliva *et al.*, 1997; Rufino *et al.*, 1997; Wintjens *et al.*, 1997; Li *et al.*, 1999). The most common criteria for classification include loop length, torsion angle conformation and type of adjacent secondary structure. In the present study, we introduce a novel algorithm for the generation of conformations as needed for prediction of loop fragments in proteins. We define loop fragments as those sections of the amino acid chain not containing secondary structure. The initialization of the look-up tables is separated from modeling and has to be executed only once. A number of look-up tables, containing loop segments of variable length, are generated and used to improve the performance in terms of both computing time and accuracy of the loop construction. The actual loop prediction is based on the so-called divide and conquer approach.

Materials and methods

Divide and conquer approach

The idea of the divide and conquer approach is to divide the loop into two segments of half the original length, choosing a good central position. These sub-segments can then be recursively divided and transformed, until the problem is small enough to be solved analytically (‘conquered’). The positions of main-chain atoms for segments of a single amino acid can be calculated analytically, using the vector representation described below. Longer loop segments can be reconstructed by geometrically transforming the coordinates for single amino acids back into the context of the initial problem. For this we need to define an unambiguous way to represent the conformation of any given residue along the chain.

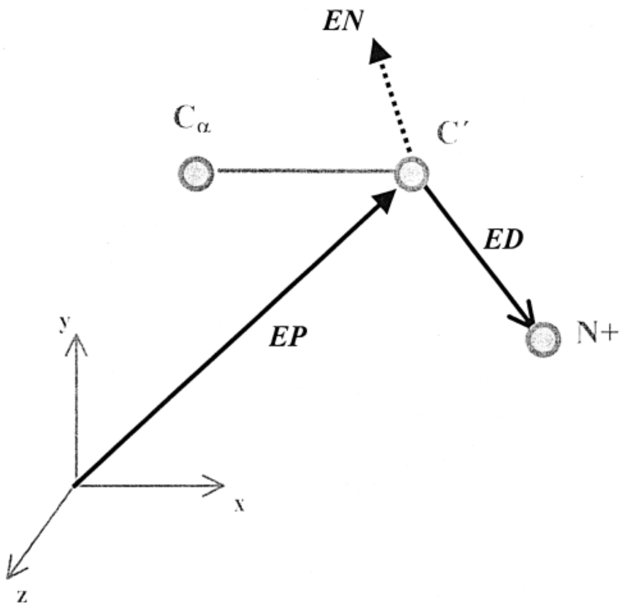

Fig. 1. Vector representation of an amino acid. The end point (EP) is the vector from an arbitrary origin to the C′ atom. The vector from the C′ atom to the N atom of the following amino acid (N+) is called the end direction (ED). The end normal (EN) vector is defined as the normal to the plane formed by the C_α, C′ and N+ atoms.

Vector representation

The conformation of an amino acid is given by the positions of its three backbone atoms, N, C_α and C′. This corresponds to three vectors, one for each atom. The absolute position of any atom in Cartesian space can also be expressed in relation to a neighboring atom. We decided to represent the conformation relative to the C′ atom. Its absolute position forms the end point (EP). The vector from C′ to the N atom of the following residue (N+) is called the end direction (ED), whereas the end normal (EN) is the normal vector of the plane defined by C_α, C′ and N+, as shown in Figure 1. Using rigid geometry, i.e. idealized values for both bond length and bond angles, this representation allows us to define the necessary operations to concatenate loop fragments, by transforming their relative orientation, to be subsequently used in the divide and conquer method.

Vector operations

Three operations have to be defined for the algorithm to work. The first operation is the re-orientation of two fragments. Let *S* and *E* be two conformations, representing the first and last residue in a loop, and *O* be the conformation of the new origin. It is possible to re-orient the whole loop relative to the new origin, such that it superimposes with *O*, using the following operations:

subtract EP_O from EP_E and EP_S
rotate ED_S to match ED_O and apply the same rotation to ED_E

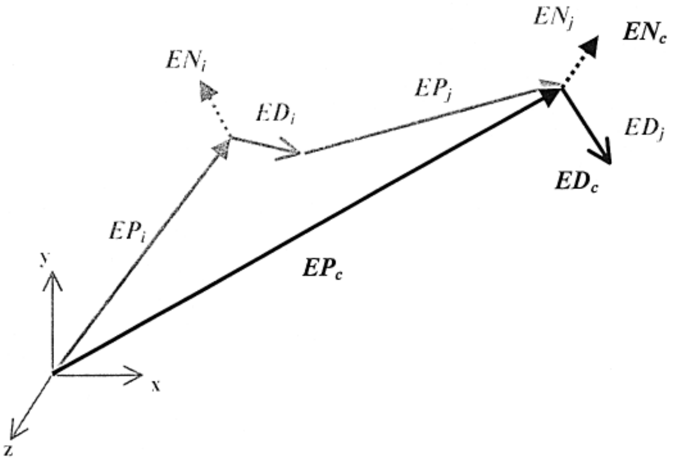

Fig. 2. A loop as described by three connected conformations in vector representation. Conformation *i* starts in the origin and spans the first half of the loop. Conformation *j* starts at the location described by conformation *i* and spans the second half of the loop. Conformation *c* describes the whole loop and is the concatenation of the other two conformations. Given any two of these conformations it is possible to calculate the third one, applying the transformations described in the text. The single vectors are described in Figure 1.

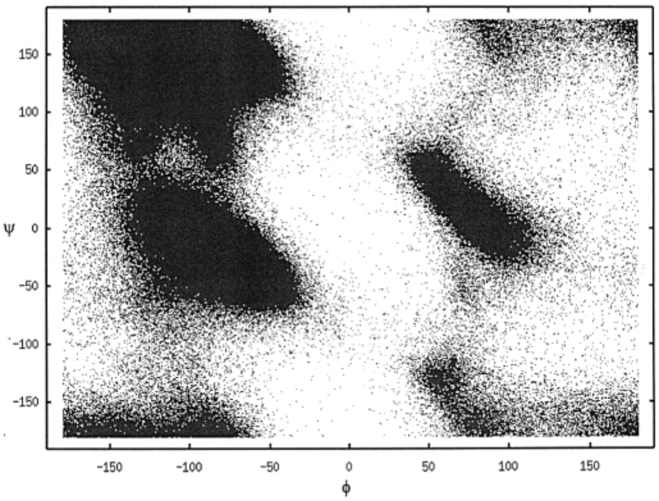

Fig. 3. Ramachandran plot of the over 1 100 000 (φ,ψ) angles, extracted from protein structures with <90% sequence identity and a resolution of 2.5 Å or better, used to create the look-up tables.

rotate EN_S to match EN_O and apply the same rotation to EN_E.

Given this operation it is possible to define the concatenation and decomposition of two loop segments. Let *i* and *j* be two loop segments, with the corresponding conformations *S* and *E* for the first and last residue in the segment. A concatenation *C* of these two segments consists of re-orienting S_j and E_j to match E_i. S_i and E_j will then represent the start and end of the concatenated loop *C*. Similarly, given the central conformation of a loop segment *i*, it is possible to decompose it in two parts, spanning origin to *i* and *i* to end, respectively. Figure 2 shows the relationship between three conformations in vector representation.

Database generation

The actual database generation requires a list of (φ,ψ) angle pairs from a Ramachandran plot (Ramachandran and Sasisekharan, 1968) distribution to be compiled. The February 2001 version of PDBSELECT 90 (Hobohm *et al.*, 1992; Hobohm and Sander, 1994) list, containing PDB identifiers with <90% sequence identity, was processed to extract the (φ,ψ) angles of loop regions. The rationale behind this high sequence cut-off being that we intended to retain as much variation in the loops with near identical sequence as possible, in order to better sample the weaker represented areas of the Ramachandran plot. In addition, only high-resolution X-ray structures solved at 2.5 Å or better were used, as lower resolution structures tend to contain more errors in the loops. The (φ,ψ) angles were computed using the DSSP (Kabsch and Sander, 1983) software and segments of regular secondary structure discarded. This reduces redundancy caused by widely populated regions of the Ramachandran plot associated with α-helices and β-sheets. The (φ,ψ) angles of over 1 100 000 residues were extracted and stored in a single table, in random order. The resulting distribution is shown in Figure 3. Whenever a new residue is considered during the subsequent database generation, different conformations are generated from these (φ,ψ) angles.

The database generation is initiated by concatenating different conformations of two single-residue fragments. Between 10 000 and 1 000 000 different conformations are generated using Monte Carlo sampling, i.e. randomly selected. Due to the random character of the process, the resulting conformations approximate the true distribution of conformations observed in protein structures. Both the end location of each segment and its central point are stored in the table using the vector representation. The central point is the overlapping residue (i.e. E_i and S_j) between the two segments from which the table entry was concatenated. It contains information for dividing the segment during database searches. The location of the starting residue (S_i) needs not be stored, as it is assumed to lie in the origin of Cartesian space. During database searches the query will thus have to be re-oriented to match this implicit starting conformation. Tables with higher order than two-residue segments are then created, starting with three residues, then four, etc. This process relies on the ability to concatenate the conformations stored in lower order tables to extrapolate longer loop segments. It is made possible by using the previously defined vector operations. Monte Carlo sampling is again used to cover conformation space in randomly selecting segments for concatenation. The process is repeated until all tables up to a chosen length have been completed. This is not limited to any specific loop length, although it can be expected that the coverage of solution space decreases for longer loops. The database represents all amino acid types. No special allowance is made for proline residues, which have a different Ramachandran distribution. This was necessary because including this difference would cause a combinatorial explosion during database generation. A filter was instead implemented to remove illegal proline conformations during the search stage.

Search algorithm

The anchor regions for the algorithm are defined as the single amino acids preceding and following the loop structure (transformed in vector representation). Using the divide and conquer approach, a loop of length *n* with an orientation *O* will be first matched against the look-up table for that length. The loop will be re-oriented to allow comparison with the database entries. Each entry contains a list of candidates from the look-up table, each with its central residue conformation. The candidate loop is divided into two loop segments of length *n*/2 (or *n*/2 + 1 and *n*/2 if *n* is an odd number). Using its central point information the loop is re-oriented and compared with a table of length $n/2$ in the following step. The process is repeated until the query conformation has reached a single residue. At this point the coordinates of the three backbone atoms can be calculated, by transforming them back into the original orientation O.

The search algorithm was designed to produce a list of possible solutions within seconds. The look-up table content is stored in a hash container sorted by the Euclidian distance D between the two anchor regions:

$$D = [\Sigma_{i\,=\,1,2,3}\,(\mathrm{EP}_i)^2]^{1/2} \qquad (1)$$

The hash container currently divides the look-up table in 64 bins. Instead of searching all entries it is possible to search only a fraction of each table, typically between 5 and 20%, to retrieve all entries with distances below a given cut-off. The search criterion, SC, for the tables is given by the distance between the target anchor region, transformed in vector representation, and each table entry. To save computing time the square-root was omitted from the formula:

$$\begin{aligned} \mathrm{SC} = {} & \lambda_{\mathrm{EP}} * \Sigma_{i\,=\,1,2,3}\,(\mathrm{EP1}_i - \mathrm{EP2}_i)^2 + \\ & \lambda_{\mathrm{ED}} * \Sigma_{i\,=\,1,2,3}\,(\mathrm{ED1}_i - \mathrm{ED2}_i)^2 + \\ & \lambda_{\mathrm{EN}} * \Sigma_{i\,=\,1,2,3}\,(\mathrm{EN1}_i - \mathrm{EN2}_i)^2 \end{aligned} \qquad (2)$$

λ_{EP}, λ_{ED} and λ_{EN} are scaling factors used to adjust the relative weight of the three vectors. λ_{ED} and λ_{EN} are generally set to 1. Increasing λ_{EP} will reduce the impact of chain orientation towards the anchor fragment, whereas reducing it will increase the propensity to select conformations with better orientation to the anchor fragment. This rule on average reduces the number of conformations to <500, or any number the user chooses. These are subjected to a number of filters, before a ranking is calculated.

The first filter relates to the geometry of the residue preceding the C-terminal anchor region. Due to the loop being constructed from the N- to C-terminus, any deviation from the idealized rigid geometry will deform the residue preceding the C-terminal anchor region. Let n be the C-terminal residue and $n - 1$ the one preceding it. The chain continuity filter, CC, checks the following conditions:

(i) bond length C'_{n-1} to N_n is 1.4 ± 0.5 Å
(ii) bond angle $C\alpha_{n-1}$, C'_{n-1} to N_n is 121° ± 15°
(iii) ω_{n-1} torsion angle is 180° ± 20°.

No allowance is made for *cis* prolines in the present implementation. The high tolerance for varations in bond length in (*i*) is due to technical reasons: Since the algorithm tends to accumulate deviations from standard geometry on the last C′–N bond length, using this high tolerance was empirically shown to preserve potentially favorable solutions. Conformations passing this filter are assumed to be close enough in rigid geometry that a constrained local optimization will be sufficient to close the gap in backbone continuity. Another filter is used to check Proline residues for approximately admissible torsion angles. A van der Waals filter (VDW) checks for inter-atomic collisions between non-bonded atoms, eliminating those conformations showing distances between two loop backbone atoms or loop and framework atoms of <2.0 Å. The following filter is based on the preference of amino acids for areas of the Ramachandran plot. The propensity P is calculated for all conformations based on the method described by Deane and Blundell (Deane and Blundell, 2000). The Ramachandran plot is divided into regions of 10×10 degrees. $P_{i,A}$, the propensity of amino acid A for area i, is calculated as follows:

$$P_{i,A} = (n_{i,A} \,/\, n_{\mathrm{tot},A}) \,/\, (n_i \,/\, n_{\mathrm{tot}}) \qquad (3)$$

$n_{i,A}$ is the number of amino acids of type A in region i and $n_{\mathrm{tot},A}$ the total number of amino acids of that type. n_i is the number of all amino acids found in region i and n_{tot} the total number of all amino acid types in the Ramachandran plot. The overall propensity for a fragment of length n is then given as:

$$V = (\Pi_{i\,=\,1,\ldots,\,n} P_{i,a}) \,/\, n \qquad (4)$$

All conformations below a given cut-off for formula (4) are removed. The propensity serves to exclude conformations for amino acids which are in a particularly unfavored region of the Ramachandran plot. Since the filters may be unable to discriminate surface loops pointing away from the protein core, a further filter was implemented as an attempt to select conformations showing a compact structure. Compactness criterion CD is calculated as the sum of the minimal distances between the C_α atoms of every residue in the loop and the protein framework C_α atoms:

$$\mathrm{CD} = \Sigma_{i\,=\,1,\ldots,\,k}\ \min_{j\,=\,1,\ldots,\,n}[(C\alpha_i - C\alpha_j)^2]^{1/2} \qquad (5)$$

Subsequently the energy E_{pot} of each fragment was calculated, using the residue-specific all-atom distance-dependent probability function (RAPDF) of Samudrala and Moult (Samudrala and Moult, 1998). E_{pot} is calculated only on the main chain and C_β atoms. Fragments with an E_{pot} score above a fixed threshold were removed.

For all remaining fragments the RMSD to the C-terminal anchor region (the N-terminal being fixed), E_{rms}, are calculated. The fragments are either ranked according to E_{rms} or a combination of E_{rms} and E_{pot}, adjusted with a scaling factor. The ranked fragments are then returned as output.

Parametrization and test sets

Because the accuracy of the predictions for different loops may vary considerably, it is desirable to parametrize and test the method on many different loops. A list including all loops from 400 non-homologous proteins (<25% sequence identity with each other) was extracted from the PDB, using random selection from the PDBSELECT25 list (Hobohm *et al.*, 1992; Hobohm and Sander, 1994). Again, only structures solved at a resolution of 2.5 Å or better were used. The regions outside regular secondary structures, as identified from evaluating the selected proteins with DSSP (Kabsch and Sander, 1983), were defined as loops for this test. Loop segments between three and 12 residues in length were selected according to the following criteria: (1) no overlap between any two loops, (2) the B factors for all main-chain atoms are <25 Å^2 and (3) the N- and C-termini are not used as test loops. This list was divided into independent parametrization and test sets. The parametrization set is composed of 200 protein structures with 777 loops in total. The test set consists of 637 loops from the remaining 200 proteins. Table I shows the distribution per loop length.

Criteria for evaluation

The accuracy of a single loop prediction is evaluated by comparing it with the native conformation. A variety of

Table I. Distribution of loops[a]

Loop length	Number of loops	
	Parametrization set	Test set
3	175	156
4	149	144
5	114	102
6	88	80
7	81	50
8	64	35
9	48	26
10	23	20
11	25	12
12	10	12

[a]The number of loops for each length in the parametrization and test sets are given.

Table II. Lowest RMSD results based on table size[a]

Loop length	RMSD (Å)		
	Large	Medium	Small
3	0.60	0.62	0.75
4	1.00	1.09	1.18
5	1.30	1.39	1.47
6	1.67	1.73	1.84
7	2.13	1.90	2.16
8	2.22	2.05	2.13
9	2.92	2.54	2.60
10	3.87	3.90	3.91
11	3.86	3.40	3.32
12	3.50	3.48	4.56

[a]The lowest average backbone (N, C_α, C′ and O atoms) RMSD results for the parametrization set are given for three different look-up table sizes: small (10 000 entries), medium (100 000 entries) and large (1 000 000 entries).

reasonable criteria for comparing loop conformations exists, with a variation of the RMSD being the most common. It is further possible to distinguish between 'local' and 'global' RMSD. The former considers a superposition of the two loops to calculate the relative internal deviation, whereas the latter superimposes the whole structure excluding the loops. It is apparent that 'local' RMSD will be lower than 'global' RMSD, as it excludes the possibility that the loop conformation may be correctly predicted, but poorly orientated to the rest of the protein. As has been argued by Fiser *et al.* (Fiser *et al.,* 2000) the two measures are correlated, with 'global' RMSD being roughly equivalent to 1.5 times 'local' RMSD. In the present paper, we have based our observations on 'global' RMS, as it is the stricter measure and also solves the optimization problem of choosing the correct orientation of the loop towards the protein framework. The actual RMSD is calculated on the N, C_α, C′ and O atoms for each residue in the loop.

Results and discussion

Performance

The quality of our algorithm was evaluated using the test set, which is composed of 1265 loops of length between three and 12 residues. We first investigated how well the algorithm was able to cover the solution space, i.e. how accurate in terms of global RMSD the best solution is. Since the look-up tables are built from a large number of (ϕ,ψ) angles, coverage is supposed to be high. However, the sampling is performed with a fixed number of entries per table, so we were interested in determining how the accuracy scales with the number of entries of the look-up table. The lowest RMSD results for various table sizes are shown in Table II. The method performs gradually better with larger look-up tables, due to the greater number of alternative loop conformations. The following tests are performed on the largest look-up table size. Computation time is found to scale linearly with loop length and the number of entries of the look-up table. It ranges between roughly 20 (three-residue loop) and 180 s (12-residue loop) for up to 200 solutions generated from tables with one million entries. For tables with 100 000 and 10 000 entries these values are, respectively, one and two orders of magnitude smaller. This linear behavior is not unexpected, since most of the time is spent searching the look-up tables. Storage of the look-up tables on hard-disk requires <38 MB per table for one million entries. Storing a database to predict loops up to 12 residues in length therefore requires <450 MB disk space. Required computer memory also scales linearly with the number of entries per look-up table. Keeping all necessary tables in memory for a given loop requires ~300 MB for loops of length 12 residues and look-up tables with one million entries. Smaller tables require ~30 MB (100 000 entries) and <5 MB (10 000 entries). Main memory requirements can be traded for computation speed by reading the tables from hard-disk during the database search.

We investigated different Ramachandran plot distributions to create the look-up tables. Alternatives included different sets of loops and an artificial distribution with Gaussian distributions approximizing main areas of (ϕ,ψ) angle space. No significant difference was encountered.

Prediction accuracy

Given the possibility to find reasonable solutions in the conformations produced by the algorithm, we were interested in defining a set of computationally inexpensive filters to reduce the alternatives for a subsequent optimization. These were fitted using the previously described parametrization set of 777 loops. Due to the way the algorithm builds the loop backbone from the fixed N-terminal anchor residue to the C-terminal anchor residue, we introduced the CC filter, which eliminates conformations lacking elementary chain continuity. For the test set this amounted to ~40%. The VDW filter was also chosen to discard conformations invalidated by strong steric clashes. The first step we intended to optimize was the selection criterium SC. This has three interdependent parameters, λ_{EP}, λ_{ED} and λ_{EN}, one for each of the three vectors. λ_{EP} differs from the other two insofar as the endpoint EP can vary the most. We tested this by using different variations of λ_{EP}, ranging from fixed values to linear and quadratic functions, to search the training set. Using a fixed $\lambda_{EP} = 0.5$ produced marginally better overall results. Changing λ_{ED} and λ_{EN} produced very similar results with no clear trend (data not shown).

The propensity filters $P_{i,A}$ and V were found to improve the accuracy, by sieving out very unlikely conformations. In order not to eliminate the possibly best solution, only conformations with impossible torsion angles for certain residues can be removed. This is in agreement with the results found by Deane and Blundell (Deane and Blundell, 2000). The compactness filter CD did not appear to improve the accuracy significantly. Therefore, it was disabled after performing initial tests.

Table III. Top *X* prediction accuracy[a]

Loop length	RMSD (Å)				
	1	3	5	10	20
3	1.06	0.89	0.80	0.74	0.69
4	1.62	1.35	1.25	1.18	1.11
5	2.22	1.75	1.62	1.45	1.38
6	2.88	2.38	2.21	1.98	1.81
7	3.62	2.65	2.52	2.37	2.21
8	3.72	2.97	2.70	2.50	2.40
9	4.95	4.12	3.75	3.22	3.01
10	6.92	5.30	4.69	4.59	4.40
11	5.88	5.40	5.12	4.15	4.10
12	6.73	5.64	4.93	4.39	4.19

[a]The average backbone (N, C_α, C′ and O atoms) RMSD over the test set for the top *X* predictions.

Table IV. Prediction accuracy for the Deane and Blundell method[a]

Loop length	RMSD (Å)	
	Top	Best
3	1.3	1.0
4	1.9	1.2
5	2.5	1.4
6	2.9	1.6
7	3.6	1.8
8	3.9	2.3

[a]The top 1 and best average backbone (N, C_α, C′ and O atoms) RMSD for the parametrization and test set from the Deane and Blundell method.

The final ranking is computed as a linear combination of E_{rms} and E_{pot}. A scaling factor is used to adjust the weight of E_{rms}. In order to optimize this factor all solutions generated for the parametrization set are stored. A complete search for the optimal value of the scaling factor has been performed. The goal was to find the scaling factor giving the lowest overall RMSD for all loops in the parametrization set. In the current implementation this is 554. Using lower scaling factors was found to increase fluctuation among the top ranking solutions. Omitting the knowledge-based potential E_{pot} from the ranking does not significantly reduce the accuracy. Therefore, filtering the solutions prior to ranking appears to make the energy function largely redundant. The optimized ranking is found to be on average a reasonable indicator for a probable loop conformation, albeit with some variations. In some cases it is possible to have several conformations with high RMSDs obscuring the best solution. Therefore, we investigated how well the native conformation would rank among the solutions. In >99.5% of the test cases the native conformation would be ranked first, and is generally ranked second in the remaining cases. Table III shows the top *X* results ($X = 1, 3, 5, 10, 20$) for the test set with the final set of filters and cut-offs. Figure 4 shows the superposition between predicted and real loops from the test set.

Comparison with existing methods

While there are a number of existing loop modeling methods, comparison is made difficult for several reasons. Different methods used for calculating the RMSD give rise to divergent results. Therefore, we have chosen to compare our results with the method of Deane and Blundell (Deane and Blundell, 2000), which is the most similar to the present method.

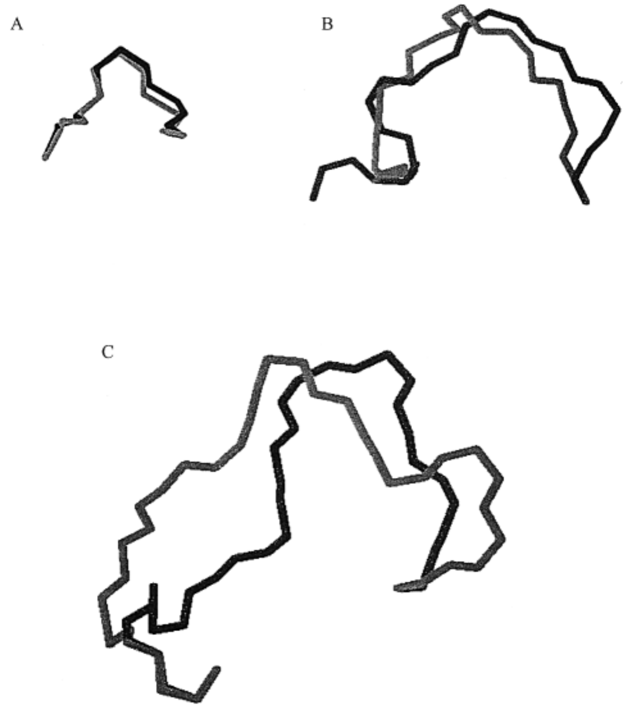

Fig. 4. Three real fragments from 1ohk, shown in dark gray. The global superposition of the best prediction is shown in light gray. The fragments represent the loop backbone N, C_α, C atoms. (**A**) Residues 172–175 (RMSD = 0.35 Å), (**B**) residues 124–130 (RMSD = 1.83 Å), (**C**) residues 77–87 (RMSD = 3.37 Å).

Table V. Performance on non-loop regions[a]

Type of structure	Top 1		DB	
	RMSD (Å)	σ_d	RMSD (Å)	σ_d
α	0.57	0.23	1.9	1.0
β	0.93	0.28	1.2	0.6
Mixture	1.51	0.93	2.6	1.5
Overall	1.22	0.83	2.2	1.4

[a]The average backbone (N, C_α, C′ and O atoms) RMSD and standard deviation (σ_d) are calculated for all overlapping length five segments of 1IGD. Top 1, the top ranking result for the present method; DB, the results from Deane and Blundell.

The method published by Deane and Blundell (Deane and Blundell, 2000) shares several similar ideas with the present work. It is also based on an algorithm for searching and ranking a database of precalculated loop conformations. Their strategy is to compute a complete enumeration of a simplified set of eight torsion angle combinations. They report an upper limit of loop length eight due to the combinatorial explosion. Our approach works with arbitrary loop lengths. Both methods make use of the same knowledge-based contact potential (Samudrala and Moult, 1998) to improve the ranking. Their method computes a set of loop conformations in the order of up to 20 min (Deane and Blundell, 2000), whereas the present work takes ~3 min on a 500 MHz PC. They use a test set of 400 high-resolution loops to validate their method. The main results for their method are summarized in Table IV. Although the test set used in this paper differs from that used by Deane and Blundell, we will nevertheless attempt to draw some conclusions. As can be seen in Table III, the present method performs better for loops up to five residues and slightly better for eight-residue loops. For loops of six and seven residues it performs as well as the one of Deane and Blundell on the top 1 solution. Longer loops cannot be compared due to length restrictions on the other method. The diminishing accuracy for longer loops can be explained considering the fixed size of our look-up tables. For long loops it becomes increasingly probable that some conformations are missed out altogether. This is supported by the performance on loops of 10 or more residues, where the average RMSD can become prohibitive.

To evaluate the prediction accuracy of the method on any fragment in a protein, we have repeated the prediction of all overlapping five-residue segments in 1IGD (immunoglobin binding protein), a small 61-residue protein containing both α-helices and β-strands. It has been already argued that this test is of particular interest to comparative modeling, where secondary structure elements are not well defined (Deane and Blundell, 2000). The results are shown in Table V.

The present method has a significantly lower RMSD than the Deane and Blundell method in all types of segments for the test protein. The most significant improvement being for α-helical and mixed segments, i.e. loops. This supports the results from the previous test for loops of length five.

Conclusion

We have presented a novel fast *ab initio* loop construction algorithm based on the divide and conquer approach. The method uses a Ramachandran plot distribution to recursively build look-up tables from the concatenation of smaller segments. This is not dependent on a particular Ramachandran plot and can predict loop segments of any size as long as a large enough number of conformations is stored in each table. In practice we can compute loops of length 12 residues with <450 MB of disk space. Memory requirements can be traded for computational speed and range between 30 and 300 MB. The search algorithm is designed to use a set of filters to allow fast computation of a ranking in a matter of seconds or up to 3 min on a desktop PC. A number of filters are implemented and several are found to be only weakly discriminating, with the chain continuity, CC, offering the largest improvement. Usage of backbone propensities also improves the results. The ranking, based on geometric and energy criteria, is sufficient to improve discrimination for short- to medium-sized loops of up to five to six residues. This is confirmed by the comparison with an existing method. The accuracy for longer loops decreases due to the limited size of the look-up tables. Usage of a more elaborate scoring function seems useful due to the variance of RMSD for high-ranking solutions. Future work will start by elucidating alternatives for the scoring function, such as non-linear ranking schemes. A protocol for improving the superposition between loop and anchor residues using local optimization should also be considered. Increasing loop flexibility by allowing variations of bond lengths and angles during database generation will also be investigated.

Due to the speed at which the method is able to produce a ranking, it may be interesting to use as a starting point for longer energy minimizations, such as those used by Van Vlijmen and Karplus (Van Vlijmen and Karplus, 1997) or Fiser *et al.* (Fiser *et al.*, 2000). This may prove to significantly reduce the computation time for such methods, allowing faster convergence than more distant starting conformations. Indeed, the method may be of use for the generation of alternative starting conformations for complex simulations, such as molecular dynamics.

Acknowledgements

We thank Achim Trabold for assistance with the test sets. We are also grateful to an anonymous reviewer for useful suggestions.

References

Abagyan,R. and Totrov,M. (1994) *J. Mol. Biol.*, **235**, 983–1002.

Abola,E.E., Sussman,J.L., Prilusky,J. and Manning,N.O. (1997) *Methods Enzymol.*, **277**, 556–571.

Brower,R.C., Vasmatzis,G., Silverman,M. and DeLisi,C. (1993) *Biopolymers*, **33**, 320–334.

Bruccoleri,R.E. (1993) *Mol. Simul.*, **10**, 151–174.

Bruccoleri,R.E. and Karplus,M. (1985) *Macromolecules*, **18**, 1767–1773.

Bruccoleri,R.E. and Karplus,M. (1987) *Biopiolymers*, **26**, 137–168.

Bruccoleri,R.E. and Karplus,M. (1990) *Biopolymers*, **29**, 1847–1862.

Bruccoleri,R.E., Haber,E. and Novotny,J. (1988) *Nature*, **335**, 564–568.

Carlacci,L. and Englander,S.W. (1993) *Biopolymers*, **33**, 1271–1286.

Carlacci,L. and Englander,S.W. (1996) *Comp. Chem.*, **17**, 1002–1012.

Chothia,C. and Lesk,A.M. (1986) *EMBO J.*, **5**, 823–826.

Chothia,C. and Lesk,A.M. (1987) *J. Mol. Biol.*, **196**, 901–917.

Chothia,C., Lesk,A.M., Tramontano,A., Levitt,M., Smith, Gill,S.J., Air,G., Sheriff,S., Padlan,E.A., Davies,D., Tulip,W.R. *et al.* (1989) *Nature*, **342**, 877–883.

Claessens,M., Cutsem,E.V., Lasters,I. and Wodak,S. (1989) *Protein Eng.*, **2**, 335–345.

Collura,V., Higo,J. and Gernier,J. (1993) *Protein Sci.*, **2**, 1502–1510.

Deane,C.M. and Blundell,T.L. (2000) *Proteins*, **40**, 135–144.

Dudek,M.J. and Scheraga,H.A. (1990) *J. Comp. Chem.*, **11**, 121–151.

Dudek,M.J., Ramnarayan,K. and Ponder,J.W. (1998) *J. Comp. Chem.*, **19**, 548–573.

Eloffsson,A., Le Grand,S. and Eisenberg,D. (1995) *Proteins*, **23**, 73–82.

Evans,J.S., Mathiowetz,A.M., Chan,S.I. and Goddard,W.A.,III (1995) *Protein Sci.*, **4**, 1203–1216.

Fidelis,K., Stern,P.S., Bacon,D. and Moult,J. (1994) *Protein Eng.*, **7**, 377–384.

Fine,R.M., Wang,H., Shenkin,P.S., Yarmush,D.L. and Levinthal,C. (1986) *Proteins*, **1**, 342–362.

Fiser,A., Kinh Giang Do,R. and Sali,A. (2000) *Protein Sci.*, **9**, 1753–1773.

Geetha,V. and Munson,P.J. (1997) *Protein Sci.*, **6**, 2538–2547.

Go,N. and Scheraga,H.A. (1970) *Macromolecules*, **3**, 178–187.

Higo,J., Collura,V. and Garnier,J. (1992) *Biopolymers*, **32**, 33–43.

Hobohm,U. and Sander,C. (1994) *Protein Sci.*, **3**, 522.

Hobohm,U., Scharf,M., Schneider,R. and Sander,C. (1992) *Protein Sci.*, **1**, 409–417.

Jones,T.A. and Thirup,S. (1986) *EMBO J.*, **5**, 819–822.

Kabsch,W. and Sander,C. (1983) *Biopolymers*, **22**, 2577–2637.

Koehl,P. and Delarue,M. (1995) *Nat. Struct. Biol.*, **2**, 163–170.

Kwasigroch,J.-M., Chomilier,J. and Mornon,J.-P. (1996) *J. Mol. Biol.*, **259**, 855–872.

Lambert,M.H. and Scheraga,H.A. (1989a) *J. Comp. Chem.*, **10**, 770–797.

Lambert,M.H. and Scheraga,H.A. (1989b) *J. Comp. Chem.*, **10**, 798–816.

Lambert,M.H. and Scheraga,H.A. (1989c) *J. Comp. Chem.*, **10**, 798–816.

Lessel,U. and Schomburg,D. (1994) *Protein Eng.*, **7**, 1175–1187.

Levitt,M. (1992) *J. Mol. Biol.*, **226**, 507–533.

Li,W., Liu,Z. and Lai,L. (1999) *Biopolymers*, **489**, 481–495.

MacKerell,J.A.D., Bashford,D., Bellott,M., Dunbrack,R.L., Evanseck,J., Field,M.J., Fischer,S., Gao,J., Guo,H., Ha,S. *et al.* (1998) *J. Phys. Chem. B*, **102**, 3586–3616.

Manocha,D., Zhu,Y. and Wright,W. (1995) *CABIOS*, **11**, 71–86.

Martin,A.C.R. and Thornton,J.M. (1996) *J. Mol. Biol.*, **263**, 800–815.

Martin,A.C.R., Cheetham,J.C. and Rees,A.R.. (1989) *Proc. Natl Acad. Sci. USA*, **86**, 9268–9272.

Morea,V., Tramontano,A., Rustici,M., Chothia,C. and Lesk,A.M. (1998) *J. Mol. Biol.*, **275**, 269–294.

Moult,J. and James,M.N.G. (1986) *Proteins*, **1**, 146–163.

Moult,J., Hubbard,T., Fidelis,K. and Pedersen,J.T. (1999) *Proteins*, **3** (Suppl), 2–6.

Nakajima,N., Higo,J. and Kidera,A. (2000) *J. Mol. Biol.*, **296**, 197–216.

Oliva,B., Bates,P.A., Querol,E., Aviles,F.X. and Sternberg,M.J.E. (1997) *J. Mol. Biol.*, **266**, 814–830.

Palmer,K.A. and Scheraga,H.A. (1991) *J. Comp. Chem.*, **12**, 505–526.

Pedersen,J. and Moult,J. (1995) *Proteins*, **23**, 454–460.
Ramachandran,G.N. and Sasisekharan,V. (1968) *Adv. Protein Chem.*, **28**, 283–437.
Rao,U. and Teeter,M.M. (1993) *Protein Eng*., **6**, 837–847.
Rapp,C.S. and Friesner,R.A. (1999) *Proteins*, **35**, 173–183.
Ring,C.S., Kneller,D.G., Langridge,R. and Cohen,F.E. (1992) *J. Mol. Biol.*, **224**, 685–699.
Rosenbach D. and Rosenfeld R. (1995) *Protein Sci*., **4**, 496–505.
Rosenfeld,R., Zheng,Q., Vajda,S. and DeLisi,C. (1993) *J. Mol. Biol.*, **234**, 515–521.
Rufino,S.D., Donate,L.E., Canard,L.H.J. and Blundell,T.L. (1997) *J. Mol. Biol.*, **267**, 352–367.
Samudrala,R. and Moult,J. (1998) *J. Mol. Biol.*, **275**, 895–916.
Shenkin,P.S., Yarmush,D.L., Fine,R.M., Wang,H. and Levinthal,C. (1987) *Biopolymers*, **26**, 2053–2085.
Sibanda,B.L. and Thornton,J.M. (1985) *Nature*, **316**, 170–174.
Smith,K.C. and Honig,B. (1994) *Proteins*, **18**, 119–132.
Sudarsanam,S., DuBose,R.F., March,C.J. and Srinivasan ,S. (1995) *Protein Sci*., **4**, 1412–1420.
Summers,N.L. and Karplus,M. (1990) *J. Mol. Biol.*, **216**, 991-1016.
Sun,Z. and Jiang,B. (1996) *J. Protein Chem.*, **15**, 675–690.
Tanner,J.J., Nell,L.J. and McCammon,J.A. (1992) *Biopolymers*, **32**, 23–32.
Thanki,N., Zeelen,J.P., Mathieu,M., Laenicke,R., Abagyan,R.A., Wierenga,R.K. and Schliebs,W. (1997) *Protein Eng*., **10**, 159–167.
Topham,C.M., McLeod,A., Eisenmenger,F., Overington,J.P., Johnson,M.S. and Blundell,T.L. (1993) *J. Mol. Biol.*, **229**, 194–220.
Tramontano,A. and Lesk,A.M. (1992) *Proteins*, **13**, 231–245.
Van Vlijmen,H.W.T. and Karplus,M. (1997) *J. Mol. Biol.*, **267**, 975–1001.
Vasmatzis,G., Brower,R.C. and DeLisi,C. (1994) *Biopolymers*, **34**, 1669–1680.
Wedemeyer,W.J. and Scheraga,H.A. (1999) *J. Comp. Chem.*, **20**, 819–844.
Wintjens,R.T., Rooman,M.J. and Wodak,S. (1997) *J. Mol. Biol.*, **255**, 235–253.
Wojcik,J., Mornon,J.P. and Chomilier,J. (1999) *J. Mol. Biol.*, **289**, 1469–1490.
Zheng,Q. and Kyle,D.J. (1994) *Proteins*, **19**, 324–329.
Zheng,Q. and Kyle,D.J. (1996) *Proteins*, **24**, 209–217.
Zheng,Q., Rosenfeld,R., Vajda,S. and DeLisi,C. (1993a) *Protein Sci.*, **2**, 1242–1248.
Zheng,Q., Rosenfeld,R., Vajda,S. and DeLisi,C. (1993b) *J. Comp. Chem.,* **14**, 556–565.
Zheng,Q., Rosenfeld,R., DeLisi,C. and Kyle,D.J. (1994) *Protein Sci*., **3**, 493–506.

Received June 8, 2001; revised December 14, 2001; accepted January 4, 2002