CABIOS

Vol. 10 no. 6 1994
Pages 587–596

Flexible algorithm for direct multiple alignment of protein structures and sequences

Adam Godzik and Jeffrey Skolnick

Abstract

The recently described equivalence between the alignment of two proteins and a conformation of a lattice chain on a two-dimensional square lattice is extended to multiple alignments. The search for the optimal multiple alignment between several proteins, which is equivalent to finding the energy minimum in the conformational space of a multi-dimensional lattice chain, is studied by the Monte Carlo approach. This method, while not deterministic, and for two-dimensional problems slower than dynamic programming, can accept arbitrary scoring functions, including non-local ones, and its speed decreases slowly with increasing number of dimensions. For the local scoring functions, the MC algorithm can also reproduce known exact solutions for the direct multiple alignments. As illustrated by examples, both for structure- and sequence-based alignments, direct multi-dimensional alignments are able to capture weak similarities between divergent families much better than ones built from pairwise alignments by a hierarchical approach.

Introduction

Protein sequence and structure alignments

Studies of similarities between protein sequences form a well-established scientific field (Waterman, 1984; Doolittle, 1990; Gribskov and Devereux 1991), with numerous applications in biology (Pearson and Miller, 1992). In a typical application, one is interested in establishing and finding the similarity between two protein sequences. The result is often expressed in terms of an alignment such as presented below as equation (1):

```
L . T P G E A Y A F
|     |     |          (1)
L S N K G   E Y S Y
```

There is a fast and deterministic algorithm to solve this problem (Needleman and Wunsch, 1970) based on dynamic programming. In this method, the best possible solution is found by iteratively extending the family of best solutions, starting from the trivial alignment of two amino acids. The computational cost of the procedure is of the order of n^2 (the square of the sequence length). Because of its speed and deterministic nature, almost all existing sequence comparison programs use this approach. This procedure works if the scoring function is local, or in other words, the gain or penalty for equivalencing amino acids at positions **i** and **j** does not depend on the alignment at other positions.

In many cases we have to use other, non-local scoring functions, for which dynamic programming does not work. For instance, if the score for the particular pair depends on another region of the alignment, then a locally optimal alignment might not be optimal after some other choice is made elsewhere. The alignment between two proteins, based on structural similarity, gives an example of a non-local scoring function. The most frequently used measure of structural similarity is the root mean square of distances between equivalent atoms in two structures after optimal superposition (RMSD) (Kabsch, 1978). The similarity measure is usually used to compare positions of the backbone Cα atoms. The scoring function for the pair $[i,j]$, which now is the distance between Cα of amino acid **i** in protein A and Cα of amino acid **j** in protein B after the optimal rotation and translation of one of the structures, depends on all equivalenced pairs in the alignment. With even small variations in the alignment, the optimal rotation might be different, resulting in different distances between every pair $[i,j]$. Therefore, at least in its standard form, the dynamic programming method cannot be used. Attempts have been made to use two-level dynamic programming to solve this type of problem (Taylor and Orengo, 1989). For every pair $[i,j]$, its score is calculated by performing conditional dynamic programming for the rest of the alignment, assuming that the first pair is already aligned. The second level dynamic programming uses a different similarity measure, which empirically was found to produce reasonable results. It is not clear if the optimal solution is in fact obtained, nor is it apparent how to generalize this approach to other scoring functions. In the absence of a fast and deterministic solution, it is possible to use other minimization approaches, such as simulated annealing, well suited to combinatorial minimization (Press *et al.*, 1993). For

Department of Molecular Biology, The Scripps Research Institute, 10666 N. Torrey Pines Road, La Jolla, CA 92037, USA.

certain non-local scoring functions—topological equivalencies in protein structures (Sali and Blundell, 1990) and the Cα-Cα distance matrix (Holm and Sander, 1993)—specialized alignment algorithms based on simulated annealing were formulated. A general alignment program capable of utilizing arbitrary scoring functions was formulated and used to compare proteins based on a variety of similarity measures (Godzik *et al.*, 1993a).

A second case when it is difficult to employ dynamic programming occurs if more than two sequences are to be aligned. It is possible to formulate the dynamic programming algorithm in more than two dimensions (Gotoh, 1986; Zuker and Somorjai, 1989; Murata, 1990; Hirosawa *et al.*, 1993). However, both the time and memory requirements soon become prohibitive, since the algorithm scales at least as n^d (n = sequence length, d = number of dimensions) both in speed and memory. Therefore, with only a few exceptions known to the authors (Murata *et al.*, 1985; Lukashin *et al.*, 1992; Hirosawa *et al.*, 1993), all multiple alignment programs use some simplifications to make the problem more tractable. These simplifications include an iterative or hierarchical pairwise alignment, scanning only some fragments of the possible alignment space or concentrating on strong, gapless fragments of the alignment (Martinez, 1988; Henneke, 1989; Higgins and Sharp, 1989; Lipman *et al.*, 1989; Vingron and Argos, 1989; Altschul and Lipman, 1990; Barton, 1990; Candresse *et al.*, 1990; Schuler *et al.*, 1991). For instance, in the hierarchical approach, all sequences are aligned pairwise with each other. The group of most similar sequences is identified on this basis, and the multiple alignment is built from the alignment within this group, by adding sequences one after the other to the previously aligned sequences. Each time a new sequence is added, it is aligned to an averaged sequence, based on the multiple alignment up to this point. However, a new sequence is not allowed to change the already existing multiple alignment. Most procedures depend on the other in which sequences are selected and require the existence of a few closely related sequences in the group. A detailed study of a full three-way alignment of the copper-binding proteins (Murata *et al.*, 1985) and virus coat proteins (Subbiah and Harrison, 1989) indicates that at least in some cases the optimal solution cannot be reached at all by an iterative hierarchical approach.

Of course, the most difficult case arises when both these problems occur simultaneously, and one is challenged to find a multiple alignment using a non-local scoring function. To the best of the authors' knowledge, no attempt to address this problem has been made to date. In this paper, a generalization of the previously presented alignment program capable of aligning two proteins (Godzik *et al.*, 1993b) to the case of the direct multiple alignment is described. The algorithm presented here makes use of the close equivalence between the alignment of proteins and the conformation of a lattice chain (Gotoh, 1986; Lipman *et al.*, 1989). Using the techniques developed in polymer physics, a general, stochastic algorithm for aligning several proteins with arbitrary scoring functions is developed.

Algorithm

Equivalence between an alignment and the conformation of a lattice chain

As the multiple alignment algorithm is a generalization of the two-way alignment, the two-dimensional case will be discussed first. First, the analogy between an alignment and a lattice chain conformation is described in considerable detail. Then we discuss the analogy between multiple alignments and lattice chain conformations in many dimensions. Finally, the dynamics of the three-dimensional lattice chain used to search the conformational/alignment space are discussed.

The two-dimensional case

The alignment between two protein sequences is in fact a set $\{[i,j]\}$ of equivalencies between positions in both sequences, where **i** denotes a position in the first and **j** in the second sequences respectively. Equivalent positions in both proteins, i.e. pairs $[i,j]$ are shown immediately above each other in equation (1). The alignment, or in other words the set $\{[i,j]\}$, can be visualized as a set of points in a plane, and this is a popular way of presenting alignments in a dot-plot alignment method (Gribskov, 1991). This set has the property that there is only one alignment point for each **i** and one for each **j**. Furthermore for two points $[i,j]$ and $[k,l]$ if $k > i$, it automatically follows that $l > j$ (this is only true for sequential alignments, i.e. those alignments where the relative order of sequence fragments is preserved). These properties are trivial as long as we remember that $\{[i,j]\}$ represents an alignment.

It is also possible to view the set of points $\{[i,j]\}$ as a particular conformation of a chain in a two-dimensional space. Various alignments between two sequences could be represented as various conformations of that chain. The process of finding the best alignment is analogous to searching for an optimal conformation of a lattice chain. This equivalence was noticed many times before (Gotoh, 1986; Lipman *et al.*, 1989) and used extensively to visualize alignments of two proteins (Gribskov and Devereux, 1991). The conformations and dynamics of conformational changes for lattice chains have been

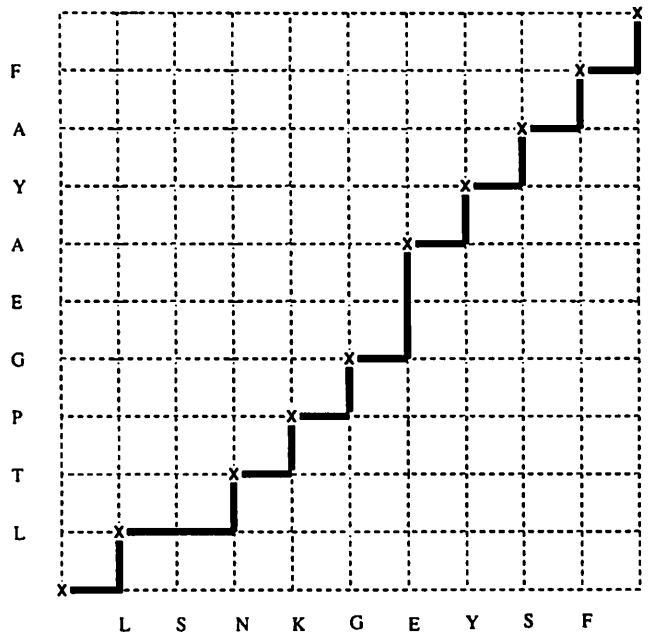

Fig. 1. The equivalence between the alignment of two proteins and a configuration of the two-dimensional lattice chain. X marks the point of the alignment (see equation 1); all other points are added according to the procedure described above.

studied extensively in polymer physics (Fixman and Stockmayer, 1970; Skolnick and Kolinski, 1990), including the dynamics of two-dimensional square (Go and Taketomi, 1978; Kolinski *et al.*, 1991) and three-dimensional cubic (Kolinski *et al.*, 1987) lattices, identical to ones used in this paper. Thus, it might be interesting to explore whether some techniques from polymer physics would be useful for the study of alignments between protein sequences.

In the simplest chain representation of an alignment, both the number of points in the chain (the number of aligned positions) and the vector length (the distances between two consecutive beads, or aligned points) might vary with the number of deletions or insertions in the alignment and with the size of the fragments which are being aligned. The study of such chains is relatively difficult, as the number of the basic vectors is very large. Therefore, a slight modification is proposed here. Points $[i,j]$ and $[k,l]$ ($k > i$ and $l > j$) from the original set are joined by vectors from the set $\{[0,1],[1,0]\}$ in such a way that the first $(k-i)$ $[1,0]$ vectors lead to a point $[k,j]$ and then $(l-j)$ vectors $[0,1]$ lead to the point $[k,l]$ (see Figure 1). With this choice, points from the original set (the aligned points) could be recognized as the only ones where a vector $[1,0]$ is immediately followed by a vector $[0,1]$; in other words, these are the only convex points along the chain. It is, of course, possible to follow a different convention for joining points from the original chain, which would result in different criteria for recognizing alignment points. There is one remaining problem: sometimes alignments do not start from the beginning of both sequences nor do they continue to the end. To ensure that the length of the chain is constant, two dummy points are introduced: $[0,0]$, where the chain starts, and $[n+1,m+1]$ where the chain ends; n and m are the lengths of the first and the second sequence respectively. The additional points in the chain do not modify the alignment, rather they are introduced solely for the purpose of simplifying the algorithm of lattice chain movement.

Now, every alignment between two protein sequences can be described in two equivalent ways. One is the traditional alignment, as shown in equation (1); the second is a particular conformation of a lattice chain built on the simple square lattice. There is a well-defined procedure for switching between the two representations of the alignment. Some obvious properties of the alignment result in restrictions on the chain conformation. Both ends are fixed, and the chain is not allowed to intersect nor to retrace its steps in any direction. In fact, it is built from a predefined number of $[1,0]$ and $[0,1]$ vectors, and both $[-1,0]$ and $[0,-1]$ vectors are not allowed. Analogous problems of self-avoiding walks (SAW) with fixed ends on a square lattice are studied in polymer physics. As mentioned above, there is a vast literature about various properties of lattice chains, including efficient algorithms for searching their conformational space.

In this analogy, the quality or score of the alignment is equivalent to the energy of a particular chain conformation. In conformational studies of the lattice models of polymer chains, traditionally the method of choice was Monte Carlo (Baumgartner, 1984), which randomly changes the conformation of the chain, and then accepts or rejects new conformations on the basis of their energy (Metropolis *et al.*, 1953). In this procedure, changes in chain conformation are introduced at random, and the energy of a new conformation is compared with that of the old. The probability of the acceptance of the new conformation is proportional to $\exp[(E_{new} - E_{old})/kT]$, where T is the temperature, which in this application is just a parameter of the simulation. Note that the configurations with lower energy are always accepted, and with increasing temperature it is increasingly easier to accept higher-energy conformations, so the system does not get trapped in local minima. A stochastic minimization does not guarantee that the lowest energy solution will be found in any given finite amount of time. However, for simple geometries, stochastic minimization methods are very efficient.

Interestingly, the analogy between the conformation of a polymer chain and sequence alignments was partly explored in the reverse direction. For some classes

of energy functions, dynamic programming has been employed to search for the minimum of the conformational energy of a polymer chain (Vajda and DeLisi, 1990).

Multiple alignments

In a multiple alignment, sets of many sequences are compared. A multiple alignment between three sequences might look like the one below.

```
IRCFITPDG  T  SKDCP
  ||          |  |
RICFNHQSSSQPQTTKTCS          (2)
  |             |||
LKCNKL     I  PIAYKTCP
```

Such an alignment can be represented by a set of equivalencies $\{[i_1, i_2, \ldots, i_d]\}$, where position i_1 from protein 1 is equivalent to the position i_2 from protein 2 and at the same time to position i_3 from protein 3, etc. **d** is the number of sequences (three in the example presented in the equation 2). In the lattice chain conformation picture of the alignment, this corresponds to a chain conformation in d-dimensional space. As before, to assure that the distance between consecutive beads remains the same, points from the original set $\{[i_1, i_2, \ldots, i_d]\}$ are jointed by vectors from the basis set, composed of **d** elementary vectors of the type $[1, 0, \ldots, 0]$, $[0, 1, 0, \ldots, 0], \ldots, [0, 0, \ldots, 1]$. The rule used to join the original points defines the convention used to recognize actual alignment points from other beads in the chain. For the three-dimensional chain, the convention where vector $[1, 0, 0]$ must be followed by the $[0, 1, 0]$ and then by $[0, 0, 1]$ was adopted. It is easy to generalize this convention to **d**-dimensions, by requiring that all **d** basis vectors should be present in **d** consecutive chain vectors in a predetermined order.

Figure 1 presented the conformational equivalent of the alignment between two proteins. (Figure 2 presents a three-dimensional lattice chain, equivalent to a multiple alignment of three proteins, shown in equation (2). The N-terminal fragment of the alignment between three toxin structures (α-cobratoxin, neurotoxin and cardiotoxin) is used as an example. Unfortunately, it is not possible to use this visualization technique to represent higher-order alignments.

For the two-dimensional case, any combination of basis vectors, other than $([0, 1], [1, 0])$, was equivalent to a deletion or insertion in the alignment and essentially all existing alignment algorithms use the same scoring system. Now, in more than two dimensions there are much more possibilities, such as aligning only **k** out of **d** sequences, and having gaps in some subset of all sequences. There are many different possible scoring systems for multiple alignments (Altschul, 1989) and essentially every program uses a different one, which makes comparisons between various programs very difficult. Here, we use one of the simpler variants (Murata *et al.*, 1985), where only positions where all sequences are aligned with no gaps contribute to the final score and every other position (with at least one gap) contributes the same value to the gap penalty. Of course, it would be perfectly possible to implement other scoring systems in the MC algorithm.

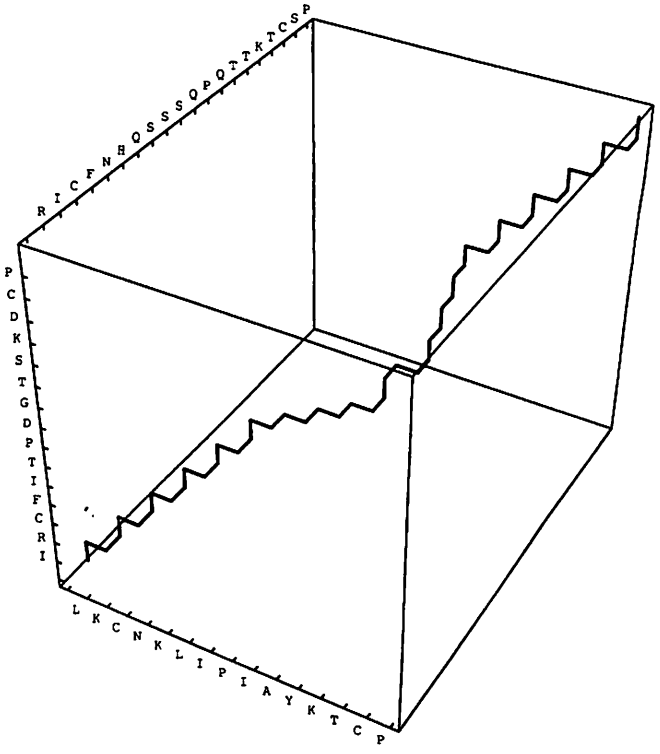

Fig. 2. The equivalence between a three-dimensional lattice chain conformation and a three-way protein alignment. This figure is equivalent to the alignment presented in equation (2).

The alignments are 'downward compatible'. Simply by removing all vectors of one type, it is possible to convert a **d**-dimensional alignment into a $\mathbf{d} - 1$ dimensional one. For instance, by removing all $[0, 1, 0]$ vectors from the three-dimensional alignment, one arrives at the two-dimensional alignment of proteins 1 and 3 from the original set. This feature of the algorithm is useful for performing two-way (or in principle any k-way) alignments within a multiple alignment. It is also possible to 'freeze' any two (or any k) dimensional 'sub-alignment', by restricting moves in such a way that no vectors in a particular two-dimensional plane are exchanged. In this way, the alignment of the chosen pair remains frozen; this

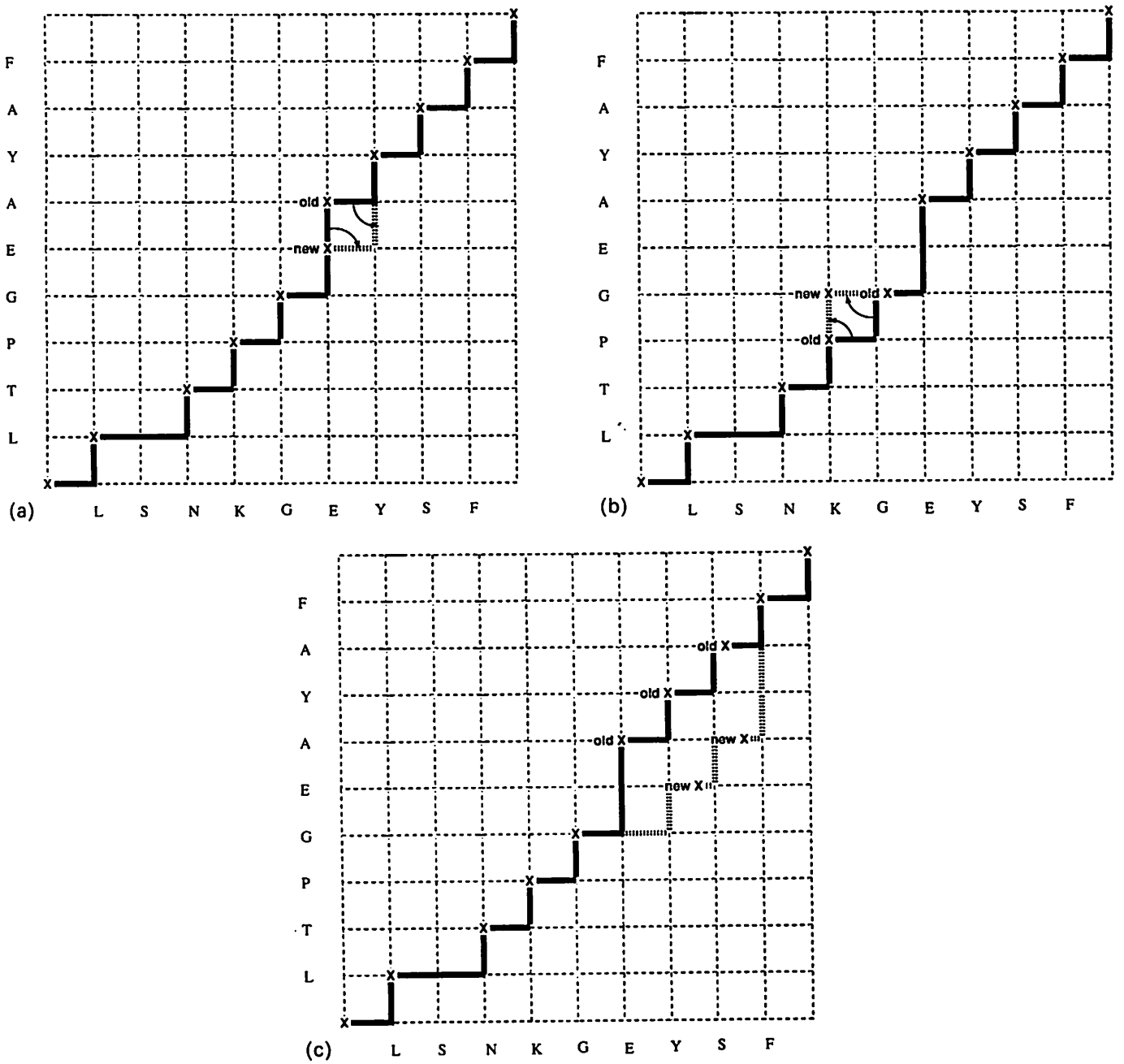

Fig. 3. Two basic local moves in the two dimensional, square lattice chain dynamics: (a) down move and (b) up move, shown here in the configurational space. (c) An example of a long-range move.

technique allows for a hierarchical build-up of the multiple alignment. Because all these changes can be done in a trivial way by restricting the set of moves used in generating new conformations of the chain, the lattice chain version of the alignment program is indeed a very flexible tool, capable of quickly changing from iterative pairwise to a true multidimensional alignment.

Again, the properties of multidimensional lattice chains have been studied in polymer physics (Fixman and Stockmayer, 1970; Skolnick and Kolinski, 1990). While the properties of polymers change dramatically with changes in space dimensions (e.g. the properties of self-avoiding chains can be solved exactly in four-, but not three-dimensional spaces de Gennes, 1979), the techniques used for studying them are essentially the same.

Dynamics of a lattice chain in three dimensions

In the two dimensions, the set of elementary moves for the MC calculations consists of exchange of the $\{[0,1],[1,0]\}$

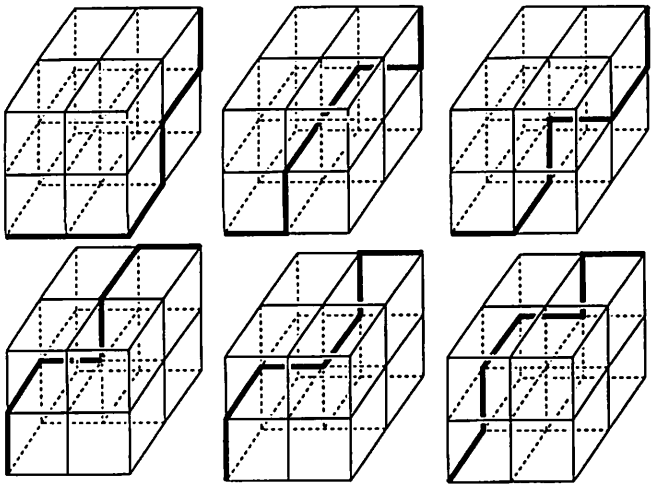

Fig. 4. Representative set of six bond moves for a three-dimensional lattice chain.

pair for the $\{[1,0],[0,1]\}$ (down move) or vice versa (up move) (Godzik, 1993). Both elementary moves are presented in Figure 3(a,b). It is possible to implement other long range moves (see Figure 3c), which are combinations of several elementary moves; however, every alignment can be reached from any other by a series of elementary moves.

In the present application, the program automatically generates the complete set of six bond moves from the set of basis vectors in a procedure that is independent of the number and type of the basis vectors. In fact, the very same program which was used to study the quality of lattice models of proteins in various lattices (Godzik *et al.*, 1993a,b) is used here. The choice of the six bond moves is quite arbitrary, since there is only a lower limit of the move length, which in three dimensions is equal to three. There is no upper limit of the move length, and six bond moves represent a good compromise between minimization efficiency and programming effort. For three dimensions, there are 729 (3^6) six vector combinations of six basis vectors. The complete library of six bond chains is built and clustered into families, based on the position of the end point. Chains within one family are equivalent in the sense that it is possible to substitute one for another, and the conformation of the chain would change only locally. Substitution of one local chain conformation for another constitutes a basic move in the MC procedure; examples of such moves are presented in Figure 4. New configurations are accepted/rejected following the Metropolis criterion.

In a single MC step, the program at random picks up a chain fragment to be changed, then cuts out the old and pastes in a new fragment. The difference in the scoring function between the old and new fragments can be calculated directly, and all calculations are performed in integer arithmetic, resulting in a very fast algorithm.

Implementation

The program utilizing the algorithm described above was implemented in FORTRAN77 on a sun SPARC10/41, but with minor changes (in particular, the random number generating subroutine) the program runs on a number of other Unix platforms (Iris, IBM 6000, HP 9000). At present, the program can handle up to three sequences, and a more general version is now under development.

Scoring functions based on the standard Dayhoff substitution matrix (Dayhoff, 1978), identity matrix and McLachlan substitution frequency matrix (McLachlan, 1971) for local sequence comparison, the RMSD between protein structures, Cα–Cα distance difference and the contact map overlap (Godzik *et al.*, 1993a,b) for structural comparisons are implemented in the program. Any other scoring function can also be incorporated as an external subroutine. A random alignment was the usual starting point, but the program has an option to start either from the previously generated alignment, or to build the initial alignment from the pairwise optimized alignments. The latter is to be preferred as it greatly enhances the rate of convergence of the Monte Carlo algorithm. In practice, it is the method of choice to reduce computational times.

In the course of the simulation, the chain conformation is modified using the set of local, six-bond moves. For each new chain conformation, the equivalent alignment is constructed and its score is calculated according to the similarity measure used in a particular run. New conformations are accepted or rejected according to the Metropolis scheme (Metropolis *et al.*, 1953), based on the energy difference between the old and the new conformation. Better conformations are always accepted, and worse ones can be accepted with a probability depending on the score difference between old and new conformations (ΔE) and $\Delta E/kT$, where T is the temperature of the system. The system was repeatedly heated and cooled, to follow a simulated annealing protocol. Heating increases the probability of accepting a higher energy conformation, and thus enables the system to escape from local energy minima. Cooling in turn speeds the approach to a local minimum. In a typical application, it was necessary to perform 10^5–10^6 elementary moves in order to achieve the convergence of the score. In all cases, the lowest score is taken to be the alignment. This corresponds to 25–30 min of CPU time on a Sun SPARC10/41 workstation. This type of minimization does not guarantee that the global extremum will be

Table 1. Maximal scores and running times (in seconds of CPU time on a Sun SPARC 10.41) for the dynamic programming and MC-based alignment programs

Family	Dynamic programming	Short MC	Long MC
Copper binding	85.0/0.3	84.4/180.0	85.0/500.0
Globins	135.7/0.3	135.0/190.0	135.7/500.0

Data were obtained with the Dayhoff substitution table without gap penalties. The score is the sum of the rescaled Dayhoff matrix as used in the GCG package (Genetic Computer Group, 1991) for all aligned pairs.

found; therefore, all alignments presented in here could possibly be improved. To get a reasonable assurance of the stability of the final alignment, multiple minimizations from different starting points were always performed. In the cases presented here, the same solution was repeatedly found. Also, for test purposes, the same MC alignment was used to obtain sequence-based alignments, where the exact solution can be found using the dynamic programming method (Murata, 1990). In all cases tested, the MC procedure converged to the unique, correct and optimal solution.

Results and discussion

At first, as a preliminary test, the algorithm was applied to the two-dimensional case with the local, sequence-based scoring function. This way, the results can be compared to the known solution, obtained by the dynamic programming.

Two cases were tested, the α- and β-chains of human hemoglobin, and the two blue copper-binding proteins, azurin and plastocyanin. In the first case, the similarity of both proteins is strong ($\sim$ 45% identical residues); in the second, it is much weaker ($\sim$ 23% identical residues). In both cases, relatively short MC runs were able to recover the alignments close to the best alignment, but much longer runs were necessary to find the unique, best alignment.

As seen from the data shown in Table I, the time requirements for the MC-based alignment program are of the order of two magnitudes larger than for the dynamic programming-based alignment. However, the time requirements for the MC program increase only slightly for such non-local scoring functions as the contact map overlap or the difference of the Cα–Cα distance maps (Godzik *et al.*, 1993a,b). Such cases cannot be treated by dynamic programming.

It is more difficult to compare results for multiple alignments, since the existing three-dimensional deterministic alignment programs based on the dynamic programming algorithm (Gotoh, 1986; Lipman *et al.*, 1989; Zuker and Somorjai, 1989; Murata, 1990; Hirosawa *et al.*, 1993) differ greatly in scoring systems used, the manner in which the penalty functions are implemented and also their availability. So here, two published alignments (Murata *et al.*, 1985; Subbiah and Harrison, 1989) and results of the iterative, pairwise multiple alignment program PileUp, as implemented in the GCG package are used for comparison (Genetics Computer Group, 1991). Note that GCG PileUp results do not give the upper bound to the score of the MC algorithm. The same groups of proteins as in original publications were used for testing: stellacyanin was added to a blue copper group to compare it to the exact solution published by Mutara *et al.*, (1985), sequences of S domains of virus coat protein from tomato bushy stunt virus (TBSV), southern bean mosaic virus (SBMV) and turnip crinkle virus (TCV) were used as a second group to compare it to the results of Subbiah and Harrison (1989).

The results in Table II illustrate two important observations. First, the time requirements of the MC algorithm grow only very slowly with the increase in the dimensionality of the alignment (cf. Table I). Second, MC alignment is able to improve upon the results of the iterative method even in the short run, and can reproduce the exact solution in the long rung. The alignment of Murata *et al.*, (1985) is reproduced exactly. Following the original publication, a different scoring function was used for virus coat proteins, and the structurally correct alignment of Subbiah is reproduced with the additional two-residue gap in the 'arm' region, preceding the S domain (see Carrington *et al.*, 1987; (Subbiah and Harrison 1989 for detailed discussion of the alignment). Note that using the same scoring function as in the case of copper proteins, the iterative approach fails completely,

Table II. Maximal scores and running times (in seconds of CPU time on a Sun SPARC 10/41) for the dynamic programming and MC-based multiple alignment programs

Family	Hierarchical approach GCG PileUp program	Short MC	Long MC	Exact solution
Copper binding	1249/0.3	1258/220.0	1271/700.0	1271
Virus coat[a]	1547/0.3	2154/210.0	2384/720.0	2360
Virus coat[b]	118/0.3	108/180.0	126/800.0	119

The GCG PileUP (GCG, 1991) program was used for comparison. PileUp and MC results were obtained either with the rescaled McLachlan substitution matrix with gap penalty equal to 12 (results denoted by[a]) or identity matrix with gap penalty equal to 4 (denoted by[b]). The exact solution for copper proteins was copied from Murata *et al.*, (1985) and for the virus coat proteins the exact solution was copied from Subbiah and Harrison (1989). Probable differences in gap penalties are responsible for the slightly better score obtained by the MC algorithm.

Table III. Maximal scores for the MC-based, three-way alignment of protein structures

Family	Pair alignment			Three-way alignment
	pair 1–2	pair 1–3	pair 2–3	
Copper binding	126/130	103/111	100/106	94
Globins	128/130	125/137	115/129	102

In the first three columns, the pairwise score from the three-way alignment is compared to the best score in separate, pairwise alignment. The data in the table were obtained with the contact map similarity measure with zero gap penalty.

while the Subbiah alignment can be 'improved', probably due to some slight differences in implementation of gap penalties.

All existing direct methods are limited so far to three sequences (Gotoh, 1986; Zuker and Somorjai, 1989; Murata, 1990; Hirosawa *et al.*, 1993) and methods which limit the space of the search to 5–10 sequence (Lipman *et al.*, 1989). As both memory and CPU requirements of MC alignments increase only slowly with number of sequences aligned, one can expect that MC alignments would be more useful in higher dimensions. We intend to extend our current algorithm in this direction in the future.

The times given in Tables I and II are for runs which started from random alignments. It is possible to use an option of the program which can start either from the dynamic programming alignment obtained using some local scoring function for the two-way alignment, or from the iterative, pairwise multiple alignment for the three-dimensional case. This approach can cut the calculation time by at least one order of magnitude.

The real advantage of the MC alignment method comes when dealing with a non-local scoring function, where it is impossible to use a dynamic programming based method. To illustrate this point, a three-way structural alignment of two different protein groups was performed (in the copper proteins group, azurin and pseudoazurin were used, as structures of stellacyanin and basic cucumber proteins are not known; globins are used as a second group, as the structure of TCV coat protein is not publicly available). The problem of comparison of three-dimensional structures of proteins is a hotly debated topic in the literature (Chothia and Lesk, 1986; Sali *et al.*, 1990; Hazes and Hol, 1992; Murzin *et al.*, 1992; Pascarella and Argos, 1992; Orengo *et al.*, 1993), with important implications for the sequence alignment field, where structural alignments are treated as a 'standard of truth'. With large families of related structures now being discovered, the ability to compare several structures simultaneously becomes increasingly important. So far, this problem has not been addressed in the literature, and either only pairwise alignments were presented and discussed, or in a way analogous to the early multiple alignments, a 'consensus' structure is defined and all structures are compared to it.

The structural alignment of three structures from our two families was done using the contact map overlap scoring function (Godzik *et al.*, 1993a,b). The algorithm converged in ~ 25 min of CPU time, with most of the slowdown due to the cost of calculating of the scoring function.

In all the cases studied, the contact map overlap was substantial, and the number of contacts common to all three structures changed little from the number common to the each pair (see Table III). This result points strongly to the conclusion that there is a set of common interactions that define a protein topology. Proving the existence of such sets and finding them for a number of topologies was the primary scientific reason behind the project that led to the development of the algorithm presented here. Also in this case, the structural multiple alignment is different from the one built from the pairwise comparison of the three structures, as seen from the first three columns of the Table III.

A standard multiple alignment in MSF format is given as output from the program. More interestingly, for structural comparisons, the program produces a superimposed contact map, which shows in detail which contacts are present in all structures and which are changed and superimposed three dimensional structures. A fragment of the superimposed contact maps is presented in Figure 5. A family of short toxin structures (the same as in Figures 1–3) is used here as an example to show the details of the contact maps in the scale of the figure.

Future developments

There are a number of possible improvements which are currently being implemented that will result in a significant speed up of the algorithm (Kolinski and Skolnick, 1987). A library of selected long-range moves is being prepared. Its implementation should significantly speed up alignments which require substantial shifts between large sequence fragments, as for instance, when a smaller protein is similar to a subdomain of a larger protein.

It is also possible to take into account the local similarity between smaller fragments (such as helices and β-strands). With little expense of computational time, the

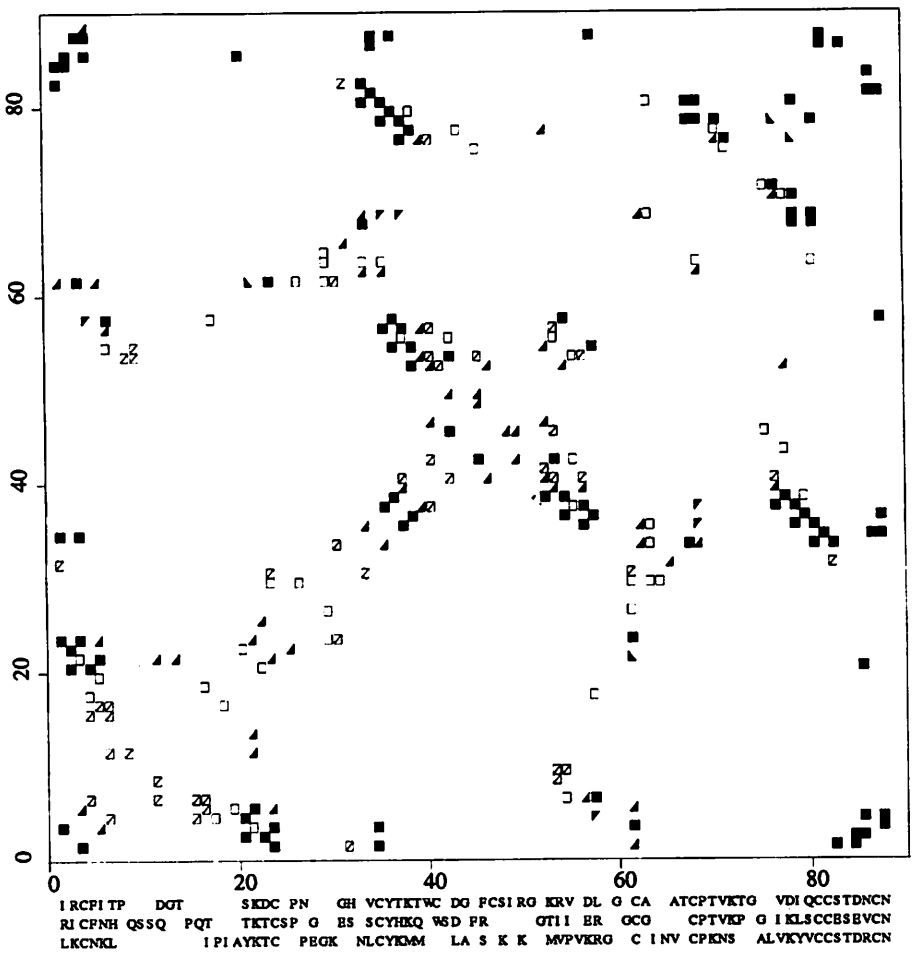

Fig. 5. Fragment of the superimposed contact map for three proteins from the toxin family. The same alignment was used in Figure 2.

whole structure can be scanned for the existence of such fragments, and subsequently, such fragments would be shifted predominantly by large-scale moves.

Acknowledgements

Stimulating discussions with Dr Andrzej Kolinski are gratefully acknowledged. This research was supported in part by grant no. P01-38794 of the Division of General Medical Sciences, the National Institutes of Health.

References

Altschul,S.F. (1989) Gap costs for multiple sequence alignment. *J. Theor. Biol.*, **138**, 297–309.

Altschul,S.F. and Lipman,D.J. (1990) Protein database searches for multiple alignments. *Proc. Natl. Acad. Sci. USA*, **87**, 5509–5511.

Barton,G.J. (1990) Protein multiple sequence alignment and flexible pattern matching. *Methods Enzymol.* **183**, 403–428.

Baumgartner,A. (1984) Simulation of polymer motion. *Annu. Rev. Phys. Chem.*, **35**, 419–435.

Candresse,T., Morch,M.D. and Dunez,J. (1990). Multiple alignment and hierarchical clustering of conserved amino acid sequences in the replication-associated proteins of plant RNA viruses. *Res. Virol.*, **141**, 315–329.

Carrington,J.C., Morris,T.J., Stockley,P.G. and Harrison,S.C. (1987) Structure and assembly of turnip crinkle turnip. *J. Mol. Biol.*, **19**, 265–276.

Chothia,C., and Lesk,A.M. (1986) The relation between the divergence of sequence and structure of proteins. *EMBO J.*, **?**, 823–826.

Dayhoff,M.O. (1978) In: Dayhoff,M.O. *Atlas of Protein Sequence and Structure*. Vol 5, Suppl. 3, pp 1–8.

de Gennes,P.G. (1979) *Scaling Concepts in Polymer Physics*. Cornell University Press, Ithaca, NY.

Doolittle,R.F. (ed.) (1990) *Molecular Evolution: Computer Analysis of Protein and Nucleic Acids Sequences. Methods in Enzymology*. Academic Press, San Diego.

Fixman,M. and Stockmayer,W.H. (1970) Polymer conformation and dynamics in solution. *Annu. Rev. Phys. Chem.*, **21**, 407–428.

Go,N. and Taketomi,H. (1978) Respective roles of short- and long-range interactions in protein folding. *Proc. Natl. Acad. Sci. USA*, **75**, 559–563.

Godzik,A., Kolinski,A. and Skolnick,J. (1993a) Lattice representation of globular proteins: how good are they? *J. Comput. Chem.*, **14**, xxx.

Godzik,A., Skolnick,J. and Kolinski,A. (1993b) Regularities in interaction patterns of globular proteins. *Prot. Eng.*, **6**, 801–810.

Gotoh,O. (1986) An improved algorithm for matching biological sequences. *J. Theor. Biol.*, **?**, 327–337.

Gribskov,M. and Devereux,J. (1991) *Sequence Analysis Primer*. Stocton Press, New York.

Genetic Computer Group (1991) *Program Manual for The GCG Package*, version 7.

Hazes,B. and Hol,W.G.J. (1992) Comparison of the hemocyanin β-barrel with other Greek key β-barrels: possible importance of the β-zipper in protein structure and folding. *Proteins*, **12**, 278–289.
Henneke,C.M. (1989) A multiple sequence alignment algorithm for homologous proteins using secondary structure information and optionally keying alignments to functionally important sites. *Compu. Applic. Biosci.*, **5**, 141–150.
Higgins,D.G. and Sharp,P.M. (1989) Fast and sensitive multiple sequence alignments on a microcomputer. *Comp. Applic. Biosci.*, **5**, 151–153.
Hirosawa,M., Hoshida,M., Ishikawa,M. and Toya,T. (1993) MASCOT: multiple alignment system for protein sequences based on three-way dynamic programming. *Comput. Applic. Biosci.*, **9**, 161–167.
Holm,L. and Sander,C. (1993) Structural alignment of globins, phycocyanins and colicin A. *FEBS Lett.*, **315**, 301–306.
Kabsch,W. (1978) A discussion of the solution of the best rotation to relate two sets of vectors. *Acta. Cryst. Allogr.*, **A34**, 827–828.
Kolinski,A., Skolnick,J. and Yaris,R. (1987) Monte Carlo studies on the long time dynamic properties of dense cubic lattice multichain systems. *J. Chem. Phys.*, **86**, 7163–7173.
Kolinski,A., Vieth,M. and Skolnick,J. (1991) Collapse of semiflexible polymers in two dimensions. Monte Carlo simulations. *Acta Phys. Pol.*, **79**, 601–612.
Kolinski,A. and Skolnick,J., (1987) Monte Carlo studies of equilibrium globular protein folding. I. Monopolymer and lattice modes of β barrel proteins. *Biopolymers*, **26**, 937–962.
Lipman,D.J., Altschul,S.F. and Kececioglu,J.d. (1989) A tool for multiple sequence alignment *Proc. Natl. Acad. Sci. USA*, **86**, 4412–4415.
Lukashin,A.V., Engelbrecht,J. and Brunak,S. (1992) Multiple alignment using simulated annealing: branch point definition in human mRNA splicing. *Nucleic Acids Res.*, **20**, 2511–2516.
Martinez,H.M. (1988) A flexible multiple sequence alignment program. *Nucleic Acids Res.*, **16**, 1683–1691.
McLachlan (1971) Test for comparing related amino acid sequences. *J. Mol. Biol.*, **61**, 409–424.
Metropolis,N.A. *et al.* (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **51**, 1087–92.
Murata,M. (1990) Three way Needleman-Wunsch algorithm. *Methods Enzymol.*, **183**,
Murata,M., Richardson,J.S. and Sussman,J.L. (1985) Simultaneous comparison of three protein structures. *Proc. Natl. Accad. Sci. USA*, **82**, 3073–3077.
Murzin,A.G., Lesk,A.M. and Cothia,C. (1992) Patterns of structure and sequence in the kunitz inhibitors interleukins-1β, 1α and fibroblast growth factors. *J. Mol. Biol.*, **223**, 531–543.
Needleman,S.B. and Wunsch,C.D. (2970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
Orengo,C.A. *et al.*, (1993). Recurring structural motifs in proteins with different functions. *Curr. Biol.*, **?**, 131–139.
Pascarella,S. and Argos,P. (1992) A data bank merging related protein structures and sequences. *Prot. Engng.*, **5**, 121–137.
Pearson,W.R. and Miller,W. (1992) Dynamic programming algorithms for biological sequence comparison. *Methods Enzymol.*, **210**, 575–601.
Press,W.H., Teukolsky,S.S., Vetterling,W.T. and Flannery,B.B. (1993) *Numerical Recipes*. Cambridge, Cambridge University Press.
Sali,A. and Blundell,T.L. (1990) Definition of general topological equivalence in protein structures. A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. *J. Mol. Biol.*, **212**, 403–428.
Sali,A., Overington,J.P., Johnson,M.S. and Blundell,T.L. (1990) From comparison of protein sequences and structures to protein modelling and design. *Trends Biochem. Sci.*, **15**, 235–240.
Schuler,G.D., Atlschul,S.F. and Lipman,D.J.P. (1991) A workbench for multiple alignment construction and analysis. *Proteins*, **9**, 180–190.
Skolnick,J. and Kolinski,A. (1990) Dynamics of dense polymer systems: computer simulations and analytic theories. *Adv. Chem. Phys.*, **77**, 223–278.
Subbiah,S. and Harrison,S.C. (1989) A method for multiple sequence alignments with gaps. *J. Mol. Biol.*, **209**, 539–548.
Taylor,W.R. and Orengo,C.A. (1989) Protein structure alignment. *J. Mol. Biol.*, **208**, 1–22.
Vajda,S. and DeLisi,C. (1990) Determining minimum energy conformations of polypeptides by dynamic programming. *Biopolymers*, **29**, 1755–1772.
Vingron,M. and Argos,P. (1989) A fast and sensitive multiple sequence alignment algorithm. *Comput. Applic. Biosci.*, **5**, 115–121.
Waterman,M.S. (1984) General methods of sequence comparison. *Bull. Math. Biol.*, **46**, 473–500.
Zuker,M. and Somorjai,R.L. (1989) The alignment of protein structures in three dimensions. *Bull. Math. Biol.*, **51**, 55–78.