Pattern-Induced Multi-sequence Alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modelling

Randall F. Smith[1] and Temple F. Smith[2]

Molecular Biology Computer Research Resource, Department of Biostatistics,
Dana-Farber Cancer Institute and School of Public Health, Harvard University
44 Binney St., Boston, MA 02115 USA

Key words: alignments / homology / modelling / protein sequence / secondary structure

ABSTRACT

A multiple sequence alignment algorithm is described that uses a dynamic programming-based pattern construction method to align a set of homologous sequences based on their common pattern of conserved sequence elements. This Pattern-Induced Multi-sequence Alignment (PIMA) algorithm can employ secondary-structure dependent gap penalties for use in comparative modelling of new sequences when the three-dimensional structure of one or more members of the same family is known. We show that the use of secondary structure information can significantly improve the accuracy of aligning structure boundaries in a set of homologous sequences even when the structure of only one member of the family is known.

---

[1] Present Address: Dept. of Cell Biology, Baylor College of Medicine, One Baylor Plaza, Houston, TX 77030 USA

[2] Present Address: Molecular Bio-Engineering Research Center, Boston University, 36 Cummington St., Boston, MA 02115 USA

INTRODUCTION

An essential step in modelling the tertiary structure of protein with unknown structure using homologous sequences of known tertiary (X-ray crystallographic) structure is the alignment of the primary sequences of the two proteins (Blundell *et al.*, 1987; Greer, 1990; Havel and Snow, 1991). Ideally, one wants to align the proteins such that there will be a one-to-one correspondence between residues that are structurally and functionally equivalent. From such an alignment, the tertiary structure can be directly modelled by assigning the known α-carbon coordinates to the equivalent residues in the sequence of unknown structure. The difficulty in aligning structures based on primary sequence information alone is that as homologous proteins diverge through the natural course of evolution, amino acid substitutions, as well as insertions and deletions can occur without affecting basic structure. As such, two distantly related proteins can have very little primary sequence similarity while maintaining striking tertiary structural correspondence. Homologous sequences do tend, however, to maintain sequence similarity within "core domains" (Chothia and Lesk, 1986; Hubbard and Blundell, 1987) and active sites (Zvelebil and Sternberg, 1988), such that it is often possible to identify structurally conserved regions (SCR's) between homologous sequences even when overall primary sequence similarity may be low (Greer, 1990).

Local optimal alignment methods can be used to identify conserved regions in a sequence of unknown structure that correspond to observed SCR's in a set of homologous sequences of known structure. The structures of these conserved regions can then be directly modelled during the initial stages of modelling an entire structure (Greer, 1990). Because they are practical and mathematically rigorous, dynamic programming-based algorithms (e.g. Needleman and Wunch, 1970;

Sellers, 1974 ; Smith and Waterman, 1981) are considered the method of choice for generating optimal alignments between a pair of proteins. These methods employ an element-to-element similarity score between paired amino acids and a gap penalty for insertions and deletions of one or more contiguous residues. They return the alignment(s) between two sequences that maximizes the overall score as a sum of the aligned element similarity scores and gap penalties.

For modelling studies, a multiple-alignment of one or more sequences of unknown structure with two or more sequences of known structure is preferable since the increased information contained in a multiple alignment can produce more precise alignments of structural domains than single paired alignments alone. While dynamic programming algorithms that perform a simultaneous multi-sequence alignment have been devised (Sankoff, 1975; Waterman *et al.* 1976; Murata *et al.*, 1985; Gotoh, 1986; Carrillo and Lipman, 1988; Altschul and Lipman, 1989; see Waterman, 1989), these methods are too computationally expensive to be of practical value when aligning more than 3-8 sequences. However, by iteratively applying pair-wise dynamic programming alignments, a wide variety of less rigorous but computationally practical methods have been developed to perform multi-sequence alignments for larger sequence sets (e.g. Barton and Sternberg, 1987a; Feng and Doolittle, 1987; Corpet, 1988; Taylor, 1988; Higgins and Sharp, 1989; Subbiah and Harrison, 1989; Vingron and Argos, 1989; Hein, 1990). Here, we describe a new multi-sequence alignment algorithm (PIMA, Pattern-Induced Multiple Alignment) based on our previously described pattern construction method (Smith and Smith, 1990) and present an extension for generating secondary structure-constrained multi-alignments for use in protein comparative modelling studies.

Secondary-structure-constrained alignments. When homologous proteins of known tertiary structure are compared, deletions/insertions are observed primarily in the unstructured coil or surface regions between secondary structure units (e.g. α-helices, β-strands) rather than within these secondary structure units (Lesk and Chothia, 1980). Apparently the packing constraints and relative rigidity of these secondary structural elements prevent internal insertions and deletions from being tolerated within these structures. One would therefore like to constrain the location of gaps when using dynamic programming to align a sequence of known tertiary structure with a homologous protein of unknown structure so that they occur primarily (or strictly) outside of aligned regions corresponding to helices and strands (Lesk *et al.*, 1986; Barton and Sternberg, 1987b) .

Fig. 1

To properly constrain gap placement during alignment, the position of gaps within both aligned sequences must be restricted. This is illustrated in Fig 1. Constraining gaps that occur within a sequence of known structure is shown in Fig 1A. During unconstrained alignment, a single position gap was placed in the middle of a β- strand (Fig 1A-1). Constraining the gap to a position outside of this β-strand, using the method described below, is shown in Fig. 1A-2. The second case, constraining gaps occurring within the sequence of unknown structure, is illustrated in Fig 1B. During unconstrained alignment (Fig 1B-1), 2 gapped regions were placed in the sequence of unknown structure. Since our underlying premise is that the basic secondary structure arrangement of two homologous sequences will be similar, given the alignment in Fig 1B-1 the second gap would occur within a secondary structure unit, removing the N-terminal portion of a β-strand. During constrained alignment, gaps within the sequence of unknown structure should be restricted so that the gap boundaries (i.e. the leading and trailing edges) are restricted to regions between secondary structure units. In Fig. 1B-2 secondary structure-dependent gap constraint generates a single gapped region with the gap boundaries placed outside of the β-strand. Below we present a method in which both of these secondary structure-constrained alignment cases are considered.

METHODS

Pattern-induced multi-sequence alignment (PIMA) algorithm. We have extended our previously described method for extracting the pattern of conserved primary sequence elements common to all members of a set of homologous proteins (Smith and Smith, 1990) to generate a multiple alignment of the sequence set during the pattern construction process: As before, given a set of related sequences, all pair-wise local optimal alignments are performed and the resulting scores are then clustered using the maximal linkage rule (Sneath and Sokal, 1973) to generate a binary dendrogram (tree); the two most similar sequences are aligned using a local optimal alignment algorithm (eq. 3, described below); a single "Amino Acid Class Covering" (AACC) pattern is constructed from the alignment by identifying, from a pre-defined amino acid class hierarchy (Fig. 2), the smallest class that "covers" the amino acids paired at each position in the alignment; patterns are constructed for each node by sequentially moving down the tree, aligning at each step the patterns/sequences connected by the next most similar node, until a single common "root" pattern is constructed for the entire tree.

Fig. 2

To generate a multi-sequence alignment, at each alignment/pattern-construction step above, the "parental" sequences are kept aligned with each "child" AACC pattern. If during subsequent alignment a gap is introduced into a pattern, a gap is inserted at the corresponding position in all of the aligned parental sequences. At the last alignment step, the final root AACC pattern will align those regions of the parental sequences that are common to the entire set. Since the root pattern can be no longer than the shortest sequence used in its construction, N- and C-terminal regions of parental sequences that extend beyond the region covered by the root pattern will only be aligned relative to their respective nodal pattern.

Dynamic programming algorithm. To describe the modified dynamic programming method used to construct secondary structure-constrained pattern-induced multiple alignments, we first review the basic dynamic programming algorithm for pairwise local optimal alignments (Smith and Waterman, 1981): Given two sequences, $A=\{A_1,A_2,A_3, \ldots A_m\}$ and $B=\{B_1,B_2,B_3, \ldots B_n\}$, a similarity matrix $H_{(m,n)}$ is generated iteratively for each $H_{i,j}$ element:

$$\begin{gathered} H_{i,j} = \max\{\, H_{i-1,j-1} + S(A_i,B_j), \\ \max_k[H_{i,j-k} - W_A(k)],\ \max_k[H_{i-k,j} - W_B(k)],\ 0\} \end{gathered} \tag{1}$$

where $H_{i,0} = H_{0,j} = 0$, $S(A_i,B_j)$ is the similarity (match) score between any two sequence elements A_i and B_j, and $W_A(k)$ and $W_B(k)$ are the gap penalties assigned to a deletion of k elements is sequences A and B, respectively, where:

$$W(k) = w_1 + w_2 * k \tag{2}$$

where w_1 and w_2 (the length-independent and length-dependent gap penalties, respectively) are constants. The longest common subsequence is determined by first locating the $H_{i,j}$ element with the maximum value. The local optimal alignment is then generated by a "traceback" procedure, sequentially determining the matrix elements leading to this maximum until an element of H equal to zero is found.

For pattern-induced multi-alignments, we employ a variant of the Smith-Waterman algorithm. Here we incorporate the "pay-once" gap penalty provisions previously described (Smith and Smith, 1990). Gapped positions created within a sequence/pattern alignment are converted into gap characters ('g') during pattern construction. During subsequent pattern alignment, the pay-once provisions (i) allow gap characters to function as either 0 or 1 amino acid of any type and (ii) eliminate the length-independent penalty (w_1) when extending a gap adjacent to a previously observed gapped region. The H matrix modified to incorporate the pay-once provisions is defined:

$$H_{i,j} = \max\{\ H_{i-1,j-1} + S(A_i,B_j),$$
$$\max_k[H_{i,j-k} - W_A(k,\ A_i,\ B_j...B_{j-k})],\ \ \max_k[H_{i,j-k} - W_B(k,\ B_j,\ A_i...A_{i-k})],\ \ 0\} \quad (3)$$

As in equation (1), the functions W_A and W_B are the gap penalties assigned to a deletion of k elements is sequences A and B, respectively. Here, however, they are a function of whether or not the deletion is across from or adjacent to one or more previously assigned gap elements:

$$W_A(k,\ A_i,\ B_j...B_{j-k}) = w_1 + w_2*k \qquad (4)$$

when A_i is not a gap character and the B_j,B_{j-k} interval does not include a gap character,

$$W_A(k,\ A_i,\ B_j...B_{j-k}) = w_2*(k-g) \qquad (5)$$

when A_i is a gap character or the B_j,B_{j-k} interval includes a gap character, where g is the number of gap characters in the B_j,B_{j-k} interval,

$$W_B(k, B_j, A_i...A_{i-k}) = w_1 + w_2*k \tag{6}$$

when B_j is not a gap character and the A_i,A_{i-k} interval does not include a gap character,

$$W_B(k, B_j, A_i...A_{i-k}) = w_2*(k-g) \tag{7}$$

when B_j is a gap character or the A_i,A_{i-k} interval includes a gap character, where g is the number of gap characters in the A_i,A_{i-k} interval.

‘

Secondary structure-dependent gap penalties. To generate secondary structure-constrained alignments, gaps within α- and 3_{10}-helices and β–strands are discouraged by increasing the gap penalty by an amount w_3 for gaps introduced within or across from (Figs. 1A and 1B, respectively) these secondary structures. To allow some flexibility in the alignment of structures, the increased gap penalty is not imposed for gap boundaries occurring at the first and last positions of a structural unit.

Consider the alignment of two primary sequences and/or patterns, A and B, where each has associated with it a co-linear sequence of secondary structure elements (as in Fig. 1), $a = \{a_1,a_2,a_3,...a_m\}$ and $b = \{b_1,b_2,b_3,...b_n\}$ (for a primary sequence with unknown secondary structure, an associated secondary structure sequence is constructed where all elements are considered unstructured [coiled]). An H matrix for which the gap penalty is increased by an amount w_3 for insertions/deletions within helices and strands can be defined by adding w_3 to equations (4) - (7), creating 4 additional conditions to equation (3):

$$W_A(k, A_i, B_j...B_{j-k}) = w_1 + w_2*k + w_3 \tag{8}$$

when A_i or B_j or B_{j-k} are within a secondary structure unit, and when A_i is not a gap character and the B_j,B_{j-k} interval does not include a gap character and ,

$$W_A(k, A_i, B_j...B_{j-k}) = w_2*(k-g) + w_3 \tag{9}$$

when A_i or B_j or B_{j-k} are within a secondary structure unit and when A_i is a gap character or the B_j,B_{j-k} interval includes a gap character, where g is the number of gap characters in the B_j,B_{j-k} interval,

$$W_B(k, B_j, A_i...A_{i-k}) = w_1 + w_2*k + w_3 \tag{10}$$

when B_j or A_i or A_{i-k} are within a secondary structure unit, and when B_j is not a gap character and the A_i,A_{i-k} interval does not include a gap character,

$$W_B(k, B_j, A_i...A_{i-k}) = w_2*(k-g) + w_3 \tag{11}$$

when B_j or A_i or A_{i-k} are within a secondary structure unit, when B_j is a gap character or the A_i,A_{i-k} interval includes a gap character, where g is the number of gap characters in the A_i,A_{i-k} interval.

Secondary structure-constrained multi-sequence alignments. When constructing a tree using the maximal linkage rule, as described above, the most similar pairs of sequences in the set, which are aligned first, may not necessarily include the sequence(s) with known structure. To constrain gap positions from the outset of the multi-alignment procedure, a "sequentially branched" tree is used to direct the order of alignments: Given a homologous set of sequences where the secondary structure is known for at least one member, one of the sequences of known structure is chosen as the "reference" sequence and is aligned to each of the other sequences in the set. The resulting set of alignment scores is sorted high-to-low and consecutively linked, producing a dendrogram with a sequentially branching topology (Fig. 3). Fig 3.

Next, to construct the multi-sequence alignment, the reference sequence is aligned with the next (the most similar) sequence in the tree using our pay-once and structure-dependent gap penalty rules described above. This alignment is then used to construct a common covering pattern. The pattern (in locked alignment with its parental primary and secondary structure sequences) is then aligned with the next most similar sequence in the tree and a new common pattern generated that aligns all of the parental primary and secondary structure sequences used up to this point in the tree. This process is iteratively repeated, moving down the tree until a root pattern is generated that is in alignment with all of the sequences in the set.

A pattern will always have at least two parental secondary structures associated with it (one from each parental sequence used in its construction). To determine the secondary structure for a pattern for the purpose of gap constraints during alignment, the following rule is used: if any one of the secondary structure elements at any given aligned position in the pattern is within a secondary structure unit (α- or 3_{10}-helix, β-sheet), then that position in the pattern is considered to be within a secondary structure unit for the purpose of gap constraints.

Identifying conserved segments within a multi-alignment. After a secondary structure-constrained multiple sequence alignment is generated, segments in the alignment with strong primary sequence similarity can be used to assign preliminary atomic coordinates to the corresponding residues in the sequences of unknown structure (Greer, 1990). To objectively and easily identify all high-similarity segments in a multi-alignment, we have developed a program that extracts all regions within the root AACC pattern of the multi-alignment that have an information density (ID) above a user-defined threshold. Here ID is the information content (IC) of a region, expressed as amino acid equivalents, divided by its length, where each contiguous gap run is treated as a single pattern element (Smith and Smith, 1990). Local regions within a pattern having an ID above a given threshold (T) can be identified using a one-dimensional dynamic programming algorithm that iteratively generates a scoring matrix, M:

$$M_i = \max \{ (M_{i-1} + I_i - T) , 0 \} \tag{12}$$

where I_i is the IC of pattern element i and T is a threshold value. The subsequence with the highest local IC is determined by first locating the M_i element with the highest value larger than 0. This position corresponds to the end of the best subsequence. By tracing back through the scoring array until the first element equal to zero is found, the beginning of the best subsequence can be identified. If j and k are the starting and ending positions of the best subsequence, the subsequence with the next highest local IC is determined by placing a value of negative infinity in elements I_j to I_{k-1} and a value of 0 in element I_k. The scoring array M is then recalculated for its entire length, and the highest greater-than-zero score will correspond to the end of the next best subsequence. All subsequences with ID above T can then be identified by repeating this process until no scores greater than 0 are identified.

Constructing test sequence sets. To test the utility of our pattern-induced multi-sequence alignment algorithm using the secondary structure-dependent gap penalty modification, we have 1) identified protein families in Brookhaven Protein Data Bank (PDB) database of protein crystal structures for which there are at least 4 non-identical members, 2) identified related sequences of unknown tertiary structure (from the SWISS-PROT protein sequence database) for each family, and 3) multi-aligned each family (now consisting of sequences of both known and unknown structure) using the secondary structure of the reference sequence to constrain the position of gaps. The Kabsch-Sander program (DSSP; Kabsch and Sander, 1983) for assigning protein secondary structure features from atomic coordinate data files in the Brookhaven Protein Data Bank (PDB) database was used to extract primary and secondary structure sequences for 488 entries in PDB rel. 19 (15 Oct. 1990), counting all subunits, if present, as separate entries. To identify sequence families, all pair-wise comparisons for the set were performed using the BLAST local alignment program (Altschul *et al.*, 1990). The sequences were then clustered using a maximal linkage rule (Sneath and Sokal, 1973) with a BLAST score cutoff of 55, generating 97 families of two sequences or more. Of these, 25 were identified as containing 4 or more non-identical sequences.

Next, the total number of individual secondary structure regions (i.e. the number of separate α- and 3_{10}-helices, β-strands, β-turns, and coiled regions) was determined for each member of each of the 25 families, and the sequence from each set with the highest number was chosen as a reference sequence. To identify related sequences of unknown structure for each family, the BLAST program was used to compare each reference sequence to all sequences in release 17 of the SWISS-PROT protein sequence database. Starting with minimum BLAST score of 95 and working towards higher scoring matches, excluding very closely related sequences in each case, 10 related SWISS-PROT sequences were chosen from each set. For those cases in which fewer than 10 matching SWISS-PROT sequences with scores above 95 were identified, sequences with scores as low as 65 were chosen in an attempt to obtain at least 10 related sequences of unknown structure for each family. The 25 families, now containing sequences with both known and unknown secondary structures, were then clustered based on their respective reference sequence, as described above.

RESULTS

To assess the degree to which secondary structure information for one or more proteins of known structure can usefully constrain the position of gaps in homologous sequences of unknown structure during multi-alignment, each of the 25 test sets were multiply aligned under three separate conditions: 1) using no secondary structure constraints; 2) using only the known secondary structure of one sequence to constrain the alignments; and 3) using the secondary structures from all of the sequences of known structure in each set to constrain the alignments. Within each of the aligned sets, all sequences with known structure were examined for the placement of gaps within, rather than between, secondary structure units (α-and 3_{10}-helices, β-strands). This was performed by first identifying all "gapped regions" in each of the alignments; a gapped region is defined as that part of an alignment covered by one or more contiguous gap characters in the root pattern (see Fig. 4). A gapped region was considered to be correctly positioned if, of those sequences of known structure containing a gap in this region, none contained a gap within a secondary structure unit. If any gap overlapped a secondary structure unit, then the entire gap region was considered to be incorrectly positioned. We chose this "all-or-none" criterion to avoid the problem of over/under-representation of the test sequences since many of the test sets contain a number of very closely related sequences.

Table 1

The number of correctly positioned gapped regions in the 25 test sets for each of the 3 alignment conditions is shown in Table 1. Since a very high secondary structure-dependent gap penalty was used ($w_3 = 10.0 * w_1$), in the case where all sequences of known structure were used to constrain gap placement (case 1) the 100% correct placement of gaps between, rather than within, secondary structure units was not unexpected. The increase from 81% to 91% correct when only a single sequence of known structure was used to constrain the alignment (combined with a corresponding decrease from 11 to 6 sets with at least one incorrect gap placement) demonstrates the utility of this approach. Structure-dependent gap constraints had little effect on the over-all quality of the the alignments, as measured by the very similar IC of the root patterns in all 3 cases.

Fig. 4

A reference sequence-constrained multi-alignment for one of the test sets (a set of viral coat protein sequences) is shown in Fig. 4A. The cluster tree for this alignment is shown in Fig. 3. The alignment contains 7 gapped regions (at reference positions 97-107, 149-153, 198, 224-234, 257-259, 290, 301-302), all of which are outside of secondary structure units. In the corresponding alignment where no secondary structure constraints were imposed, 2 of the 7 gapped regions overlapped a secondary structure unit boundary. One of these regions (between positions 91-113) is shown in Fig 4B (the corresponding region is boxed in Fig

4A). In Fig. 4B, the gapped region in the last sequence in the set (POLG$BOVEV) overlaps the last two residues of the leading β-strand in the PDB sequences 2RLT-1, 1RMT-1, and 2RT5-1, and thus was marked as an incorrectly aligned region. In boxed region in Fig. 4A, the gapped region in the POLG$BOLVEV sequence is the same length as in Fig. 4B, but it is shifted to the right by one residue, overlapping the β-strand by an allowable single residue.

DISCUSSION

Our multiple alignment algorithm incorporates several unique features: 1) it is based on a local dynamic programming algorithm; 2) alignments are based on the pattern of conserved and non-conserved elements/domains common to a set of homologous sequences; 3) gaps within a multiple alignment are represented in a simple and consistent manner; and 4) secondary structure-dependent gap penalties are imposed both within and across from known secondary structures. First, our method employs a local (Smith and Waterman, 1981), rather than global (Needleman and Wunch, 1970, Sellers, 1974) dynamic programming algorithm. While a local optimal alignment is guaranteed to find the maximally similar subsequence(s) above a set similarity threshold, global alignment procedures, which align the entire length of one sequence with the entire length of another, have no such guarantee. Thus in cases where a set of sequences share only a single short conserved domain (such as a zinc finger domain), the local-based multiple alignment method should be better able to properly align a sequence set on such a domain.

Our multiple alignment method, which employs our previously described pattern-construction method (Smith and Smith, 1990), aligns those elements/domains that are conserved amongst all of the sequences under alignment. After each alignment/pattern-construction step during multiple alignment, only the conserved sequence elements are carried forward to the next alignment step; non-conserved positions are converted into wild-card (’X’) and gap (’g’) elements that act as spacers between conserved elements/regions. Thus during subsequent alignment, conserved domains are more likely to align with their homologous regions in other sequences/patterns, and non-conserved regions are less likely to align spuriously with non-homologous regions.

The pay-once gap penalty rule encourages gaps to be placed in contiguous units, both horizontally (within a sequence) and vertically (at the same aligned position in homologous sequences) within a multi-sequence alignment, rather than being fragmented. Gaps within a multiple alignment are represented and handled in a simple and consistent manner, where each gap character in a gapped region can behave as either 0 or 1 amino acid during subsequent alignment. This simple gap representation scheme alleviates the algorithmic complications associated with assigning pairwise gap costs to multiple alignments (Altschul, 1989).

Dynamic programming algorithms that employ secondary structure-dependent gap penalties during alignment have been previously described for pair-wise alignments (Lesk *et al.*, 1986; Barton and Sternberg, 1987b) and have been adopted for multi-sequence alignment (Henneke, 1989). Profile-based alignment methods, where a set of pre-aligned sequences is aligned with a single sequence using a position-specific scoring table, also employ position- (including secondary structure-) dependent gap penalties (Gribskov *et al.* 1987, 1990; Luthy *et al.*, 1991; Bowie *et* *al.*, 1991). However, from the descriptions of these methods, gaps appear only to be constrained when they occur within secondary structure units (helices or sheets) in sequences of known structure (Fig. 1A); gaps do not appear to be constrained when they occur across from secondary structure units (Fig. 1B). Constraining gaps only when they occur within the sequence of known structure is not sufficient to produce correctly constrained alignments across an entire set, as shown in Fig.1.

In our method and in recent profile-based alignment studies (Luthy *et al.*, 1991; Bowie *et al.*, 1991), the gap penalty is *increased* for gaps placed within secondary structure units, and the normal penalty is applied to gaps placed between structures. In contrast, in the methods of Lesk et al. (1986), Barton and Sternberg (1987b) and Henneke (1989), the gap penalty is *reduced* for gaps placed between secondary structure units, and the normal gap penalty is applied to gaps placed within structures. Our rationale for choosing the former method is to restrict the placement of gaps within helices and sheets rather than to encourage the placement of gaps between these structures. We believe this to be more biologically meaningful and, in addition, it complies with the general dynamic programming rule that the gap penalty be greater than the maximum amino-acid-to-amino-acid match score when insertion/deletion events are expected to be rarer than substitutions (Fitch and Smith, 1983). Decreasing the normal gap penalty for gap placement between structures would violate this rule for most similarity matrices.

The methods described here should prove useful in comparative modelling studies. Given a set of homologues in which the 3D-structure is known for one or more members, a multiple alignment can be performed using our pattern-induced and structure-constrained alignment method. The multiple alignment can then be used directly to model the structure of the homologues of unknown structure (Havel and

Snow, 1991). Alternately, the method described here to extract high-ID subsequences from the root AACC pattern can be used to identify the most highly conserved regions within the alignment (e.g. the regions bold type in Fig. 4). In these conserved regions, tertiary structural information from the homologues of known structure can be used to assign preliminary atomic coordinates to the corresponding residues in the sequences of unknown structure as the initial stage in modelling the entire structure (Blundell *et al.*, 1987; Greer, 1990).

Our secondary structure-constrained multi-sequence alignment method should also prove useful for variety of other types of studies. For example, this method could be used to create multi-sequence alignments for use in constructing sequence profiles and patterns (e.g. Taylor, 1986; Bashford *et al.*, 1987; Gribskov *et al.*, 1987, 1990; Lathrop *et al.*, 1987; Patthy, 1987; Staden, 1988; Barton and Sternberg, 1990, Bowie *et al.*, 1990; Sibald and Argos, 1990; Smith and Smith, 1990), especially when a sequence set to be aligned contains one or more sequences of known 3D-structure. Our secondary structure-constrained alignment algorithm could also be used in those methods that employ dynamic programming to align profiles and patterns with other sequences during database searches.

Sander and Schneider (1991) describe the construction of a database of ’homology-derived protein structures’ in which all sequences of known structure are multiply aligned with their homologs of unknown structure. Our secondary structure-dependent gap penalty algorithm could be used to improve the structural alignments of these sequences sets, which would be particularly useful when using such sets to generate statistical parameters (e.g. Chou and Fasman, 1974a,b; Garnier *et al.*, 1978; Richardson and Richardson, 1988; see Fasman, 1989) and sequence motifs (e.g. Lim, 1974; Cohen *et al.*, 1986; Rooman and Wodak, 1988; Overington *et al.*,

1990) for structure prediction. In addition, our method for identifying high-ID subsequences could be used to recognize regions that are likely to be functionally conserved within proteins, such as active sites (Zvelebil and Sternberg, 1988). Specific residues within such regions could then be targeted for site-specific mutagenesis studies.

[small type?]

Software Availability. The software described here is available free of charge to non-profit organizations (a distribution fee and non-resale agreement is required for commercial use). The current implementation runs only under the UNIX operating system. Copies can be obtained 1) electronically via INTERNET anonymous FTP to mbcrr.harvard.edu; the package is in a single compressed tar file in the ’MBCRR-Package’ sub-directory, or 2) by sending us a Sun or SGI 1/4" cartridge tape.

ACKNOWLEDGEMENTS

We thank our colleagues Karen Kabnick, Steen Knudsen, and Collin Stultz for critical reading of the manuscript.

This work was supported by Public Health Service Grant RR02275 from the National Institutes of Health and Grant DIR-8715633 from the National Science Foundation.

REFERENCES

Altschul, S.F. (1989) J. Theor. Biol. **138**, 297-309.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. (1990) J. Mol. Biol. **215**, 403-410.

Altschul, S.F. and Lipman, D.J. (1989) Siam J. Appl. Math. **49**, 197-209.

Barton, G.J. and Sternberg, M.J.E (1987a) J. Mol. Biol. **198**, 327-337.

Barton, G.J. and Sternberg, M.J.E (1987b) Protein Eng. **1**, 89-94.

Barton, G.J. and Sternberg, M.J.E (1990) J. Mol. Biol. **212**, 389-402.

Bashford, D., Chothia, C., and Lesk, A.M. (1987) J. Mol. Biol. **196**, 199-216.

Blundell, T.L., Sibanda, B.L., Sternberg, M.J.E., and Thorton, J.M. (1987) Nature **326**, 347-352.

Bowie, J.U., Luthy, R., and Eisenberg, D. (1991) Science **253**, 164-170.

Bowie, J.U., Clarke, N.D., Pabo, C.O, and Sauer, R.T. (1990) Proteins **7**, 257-264.

Carrillo, H. and Lipman, D. (1988) Siam J. Appl. Math. **48**, 1073-1082.

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

Chou, P.Y.and Fasman, G.D. (1974) Biochemistry **13**, 211-222.

Chou, P.Y.and Fasman, G.D. (1974) Biochemistry **13**, 222-245.

Cohen, F.E., Abarbanel, R.M., Kuntz, I.D., and Fletterick, R.J. (1986) Biochemistry **25**, 266-275.

Corpet, F. (1988) Nucl. Acids Res. **16**, 10881-10890.

Fasman, G.D. (1989). In Fasman, G.D. (ed.) *Prediction of Protein Structure and the Principles of Protein Conformation.* Plenum Press, New York, pp. 193-316.

Feng, D.-F. and Doolittle, R.F. (1987) J. Mol. Evol. **25**, 351-360.

Fitch, W.M. and Smith, T.F. (1983) Proc. Natl. Acad. Sci. USA **80**, 1382-1386.

Garnier, J., Osguthorpe, D.J., and Robson, B. (1978) J. Mol. Biol. **120**, 97-120.

Gotoh, O. (1986) J. Theor. Biol. **121**, 327-337.

Greer J. (1990) Proteins **7**, 317-334.

Gribskov, M., McLachlan, A.D., and Eisenberg, D. (1987) Proc. Natl. Acad. Sci. USA **84**,4355-4358.

Gribskov, M., Luthy, R., and Eisenberg, D. (1990) Methods Enzymol. **183**, 146-159.

Havel, T.F. and Snow, M.E. (1991) J. Mol. Biol. **217,** 1-7.

Henneke, C.M. (1989) CABIOS **5**, 141-150.

Hein, J. (1990) Meth. Enzymol. **183**, 626-645.

Higgins, D.G. and Sharp, P.M. (1989) CABIOS **5**, 151-153.

Hubbard T.J.P. and Blundell, T.L. (1987) Protein Eng. **1**, 159-171.

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

Lathrop, R.H., Webster, T.A., and Smith, T.F. (1987) Commun. ACM **30**, 909-921.

Lesk, A.M. and Chothia, C. (1980) J. Mol. Biol. **136**, 225-270.

Lesk, A.M., Levitt, M., and Chothia, C. (1986) Protein Eng. **2**, 77-86.

Lim, V.I. (1974) J. Mol. Biol. **88**, 873-894.

Luthy, R., McLachlan, A.D., and Eisenberg, D. (1991) Proteins **10**, 229-239.

Murata, M., Richardson, J.S., and Sussman, J.L. (1985) Proc. Natl. Acad. Sci. USA **82**, 3073-3077.

Needleman, S.B. and Wunch, C.D. (1970) J. Mol. Biol. **48**, 443-453.

Overington, J., Johnson, M.S., Sali, A., and Blundell, T.L. (1990) Proc. R. Soc. Lond. B **241**, 132-145.

Patthy, L. (1987) J. Mol. Biol. **198**, 567-577.

Richardson, J.S., and Richardson, D.C. (1988) Science **240**, 1648-1652.

Rooman, M., and Wodak, S.J. (1988) Nature **335**, 45-49.

Sander, C. and Schneider, R. (1991) Proteins **9**, 56-68.

Sankoff, D. (1975) SIAM J. Appl. Math. **28**, 35-42.

Sellers, P.H. (1974) SIAM J. Appl. Math. **26**, 787-793.

Sibald, P.R. and Argos, P. (1990) CABIOS **6**, 279-288.

Smith, R.F. and Smith, T.F. (1990) Proc. Natl. Acad. Sci. USA **87**, 118-122.

Smith, T.F. and Waterman, M.S. (1981) J. Mol. Biol. **147**, 195-197.

Sneath, P.H. and Sokal, R.R. (1973) *Numerical Taxonomy* (Freeman, San Francisco).

Staden, R. (1988) CABIOS **4**, 53-60

Subbiah, S. and Harrison, S.C. (1989) J. Mol. Biol. **209**, 539-548.

Taylor W.R. (1988) J. Mol. Evol. **28**, 161-169.

Vingron, M. and Argos, P. (1989) CABIOS **5**, 115-121.

Waterman, M.S. (1989) In Waterman, M.S. (ed.), *Mathematical Methods for DNA Sequences.* CRC Press, Boca Raton, Florida, pp. 53-92.

Waterman, M.S., Smith, T.F., and Beyer, W.A. (1976) Adv. Math. **20**, 367-387.

Zvelebil, M.J.J.M and Sternberg, M.J.E (1988) Protein Eng. **2**, 127-138.

Table 1. Number of correctly positioned gap regions in the 25 test sets when aligned under different gap constraint conditions.

Gap constraints[a] using:	No. gapped regions	No. correct		Avg. pattern IC[b]	Avg. pattern ID[c]
1- all seqs w/ 3D-struct	101	101	(100%)	54.3	0.33
2- reference seq only	103	94	(91%)	54.4	0.33
3- no constraints	105	85	(81%)	54.6	0.33

Avg, average; seq(s), sequence(s); struct, structure.

[a] Gap penalties employed: w_1=6.67, w_2=1.33, w_3= 66.7

[b] Pattern information content is expressed as amino acid equivalents (Smith and Smith, 1990).

[c] Pattern information density = (pattern IC)/(pattern length); each run of contiguous gap characters is treated as a single element in the calculation of pattern length.

FIGURE LEGENDS

Fig. 1. Illustration of unconstrained (1) and secondary structure-constrained (2) gap placement during alignment . Top sequence of A and B, segment of a sequence of known structure. Middle sequence of A and B, secondary structure assignments for the above sequence: e, β-strand; t, β-turn; c, coil. Bottom sequence of A and B, homologous sequence of unknown structure from the SWISS-PROT database. (A) Insertion of a gap within the sequence of known structure. (B) Insertion of gap(s) within the sequence of unknown structure.

Fig. 2. The amino acid class hierarchy used to construct AACC patterns during multi-alignment. Uppercase characters, one-letter amino acid codes; lowercase characters, designated amino acid classes; X, wild-card character representing one amino acid of any type. The match score between any two aligned elements equals the score assigned to the minimally inclusive class in the hierarchy that includes both elements.

Fig. 3. Cluster dendrogram for the sequence set shown in Fig. 4.

Fig. 4. Pattern-induced multiple sequence alignment of one of the 25 test sets described in the text (test set 14, a family of viral coat protein sequences). For space considerations, very closely related homologues are not shown (PDB loci 4RHV_1, 2RS3_1, 2RS1_1, 2RR1_1, 2RM2_1, 2R07_1, 2R06_1, 2R04_1, 1R08_1, and 2RMU_1; SWISS-PROT loci POLG$COXB3, POLG$HRV14, and POLG$HRV1B) (A) Secondary structure-constrained multi-alignment using structural information from the reference sequence only (sequence 2: PDB locus 2PLV_1). The root AACC sequence pattern generated from this set is shown as

sequence 1; amino acid classes (lowercase characters) are defined as in Fig. 2; X, wild-card character (one amino acid of any type); g, gap character (zero or one amino acid of any type). Secondary-structure assignments for sequences with know structure are shown under the corresponding PDB primary sequence and are labeled with the suffix '.ss': h, α-helix; e, β-strand; g, 3_{10}-helix; t, β-turn; c, coil. The boxed region (at position 91-113) corresponds to the region shown in B. Information-dense regions (identified within the root AACC pattern using equation (12) and an ID-threshold of 0.45 amino acid equivalents) are shown in bold type. (B) Multi-alignment of the same set using no secondary-structure constraints; only one region of the alignment is shown (reference positions 91-113, corresponding to the boxed region in A).