Pacific Symposium on Biocomputing 6:275-286 (2001)

A NEW ALGORITHM FOR THE ALIGNMENT OF MULTIPLE PROTEIN STRUCTURES USING MONTE CARLO OPTIMIZATION

C. GUDA, E. D. SCHEEFF, P. E. BOURNE[1,2], I. N. SHINDYALOV
San Diego Supercomputer Center, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0537
[1]Department of Pharmacology, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093
[2]The Burnham Institute, 10901 North Torrey Pines Road, La Jolla, CA 92037

We have developed a new algorithm for the alignment of multiple protein structures based on a Monte Carlo optimization technique. The algorithm uses pair-wise structural alignments as a starting point. Four different types of moves were designed to generate random changes in the alignment. A distance-based score is calculated for each trial move and moves are accepted or rejected based on the improvement in the alignment score until the alignment is converged. Initial tests on 66 protein structural families show promising results, the score increases by 69% on average. The increase in score is accompanied by an increase (12%) in the number of residue positions incorporated into the alignment. Two specific families, protein kinases and aspartic proteinases were tested and compared against curated alignments from HOMSTRAD and manual alignments. This algorithm has improved the overall number of aligned residues while preserving key catalytic residues. Further refinement of the method and its application to generate multiple alignments for all protein families in the PDB, is currently in progress.

1 Introduction

Many algorithms have been developed for the pair-wise alignment of protein structures[1-4]. However, few efficient approaches are available for obtaining the alignment of multiple structures[5-7]. Rapid advances in experimental techniques have resulted in determination of more than 12000 protein structures to date and the rate of growth in structural information is expected to rise further in the era of structural genomics. A global and comprehensive study of protein structures is possible only by comparison of multiple structures and investigation of their folding similarities and evolutionary relationships. With the availability of vast amounts of structural information, accurate and fully automated structural alignment algorithms are needed for a better understanding of sequence-structure-function relationships in proteins. Here, we present a new algorithm for the alignment of multiple protein structures using Monte Carlo optimization method.

2 Methods

2.1 Data Preparation

Input data were taken from an all-to-all structure alignment database produced using the Combinatorial Extension (CE) algorithm for pair-wise structural alignment[3] implemented using the Property Object Model (POM) data management system[8]. The all-to-all structure alignment database contains pair-wise alignment data on all

against all comparisons between the representative (non-redundant) structures from PDB. Initial (zero-approximation) multiple alignments were produced by assembling pair-wise alignment data with respect to the selected master structure. Average RMSD values were calculated for each structure against all other structures based on C_α - C_α inter residue distances and the structure with minimum average RMSD was chosen to be the new master. Member structures were superposed with the new master structure. The alignment data so obtained were used as a starting point for optimization.

2. 2 Terminology

The multiple alignment is represented by two distinct types of regions, the alignment 'block' (Fig. 1A) and the 'free pool' (Fig. 1B) of unaligned residues. These two types of regions are defined and regulated by two parameters, minimum block length (L_{min}) and minimum number of residues in a given alignment column (R_{min}). The alignment block is a contiguous region of alignment columns (Fig. 1C) with a length $L \geq L_{min}$ and each aligned column has residues $R \geq R_{min}$. R_{min} is determined by a cutoff percentage of the number of structures being aligned. In our studies, L_{min} = 4 and R_{min} = 33% of the number of structures in a given alignment. The region separating two adjacent aligned blocks is the free pool region. Each row corresponds to one protein structure in the multiple alignment (Fig. 1D).

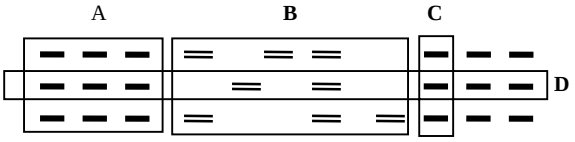

Figure 1. Black strips denote residues in alignment blocks (aligned residues), double-line strips denote free pool (unaligned) residues. A – block. B - free pool, C – alignment column, D – alignment row.

2. 3 Algorithm

The goal of a multiple alignment algorithm is to increase the number of alignment columns and consequently reduce the number of residues in the free pool region, within the reasonable limits of an alignment distance change. We used the Monte Carlo (MC) approach to achieve this. We have designed four types of moves (Figure 2) and an appropriate scoring function.

2. 3. 1 Scoring function

A distance-based score was calculated for each column in the alignment block. Geometric distances were calculated from the 3-D coordinates of C_α atoms for

each pair of residues in a column for $R(R-1)/2$ combinations, where R is the number of residues in a column. Column distances were defined as average geometric distances calculated for each column. The alignment score S was calculated (similar to Gernstein and Levitt, 1998)[9] from the column distances in aligned blocks, using the following scoring function:

$$S = \sum_{i=0}^{l} \left[\frac{M}{1+(d_i/d_0)^2} - A \right] - G \qquad (1)$$

where, l is the total number of aligned columns, M = 20 is the maximum score of a match, d_i is the average distance for column i, d_0 is the maximum distance which is not penalized and $A = \begin{cases} 0, \text{if } d_i \le d_0 \\ 10, \textit{if } d_i > d_0 \end{cases}$. The value of d_0 is chosen from the initial distance distribution for all columns in the multiple alignment at the boundary for the top 10% of column distances. G is linear gap penalty term with gap initiation and gap extension penalties of 15 and 7, respectively.

2. 3. 2 Move set

Four types of moves, each to be applied in a forward and backward direction, were designed (modified from Mirny and Shakhnovich, 1998)[4] to address different alignment situations. The types of moves are: (i) The 'shift' move shifts residues in one randomly chosen structure (Fig. 2A); (ii) the 'expand' move expands the alignment block by acquiring newly aligned residues from the free pool (Fig. 2B); (iii) the 'shrink' move shrinks the blocks by pushing the outer column residues into the free pool (Fig. 2C); (iv) the "split and shrink' move splits longer blocks in two and shrinks one of the new blocks (Fig. 2D).

In the course of optimization the type and position of each move are selected randomly.

2. 3. 3 Search space and search constraints

(i) Shift: Residues of an individual structure in a block can be shifted in a forward or backward direction, when there are residues available in the free pool at the opposite direction of the shift. The structure to be shifted is chosen randomly, the fragment is shifted one position at a time and the length of shift is from 1 to n where, n is the number of residues available in the free pool for a given row.

(ii) Expand: Expansion of aligned blocks is possible when there are enough residues in the free pool in the direction of the move to fill the newly acquired column with $R \geq R_{min}$ residues. Aligned blocks are expanded one column at a time as long as

eligible residues are available from the free pool without changing the residue order in the alignment.

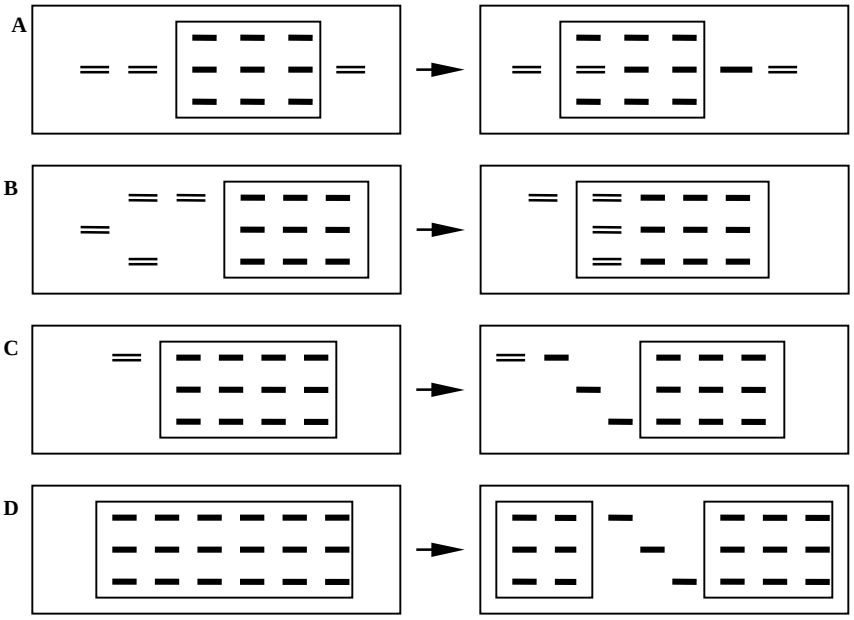

Figure 2. Move set. Left and right columns give alignment before and after move respectively. Black and double-line strips denote residues in alignment blocks and free pool (respectively) before the move. Location of alignment blocks shown in boxes. A - shift, B - expand, C - shrink, D - split and shrink.

(iii) Shrink: Aligned blocks can be shrunk in either direction resulting in pushing the residues from the outer aligned columns into the free pool.

(iv) Split and Shrink: Longer aligned blocks with a length $L \geq 3L_{min}$ can be split randomly into two sub-blocks with each sub-block of length $L \geq L_{min}$. The longer of the sub-blocks can be shrunk towards the inside of the split until its length is $L' \geq 1$.

3 Results and Discussion

We have selected 66 families from representative structural families in the CE all-to-all database[10]. Only families with a certain number of structures N in the family were considered, where $30 \geq N \geq 8$. The selection of structural neighbors was made according to a CE z-score value $z \geq 4.0$, and the slave structures containing C_α coordinates for at least 50% of the residues that are in alignment with the master structure. The MC algorithm was run against all the 66 families at different

parameter settings. We also generated multiple alignments for two specific families, the protein kinases and the aspartic proteinases and compared the results to unmodified CE pairwise alignments and alignments available from HOMSTRAD[11].

3.1 Selection of the optimization protocol

The main principle of MC optimization is iterative improvement through a random walk of the search space, with occasional excursions into non-optimal territory. Specifically, location and move type are selected randomly and the change in the alignment score (ΔS) is evaluated. If $\Delta S > 0$, the move is always accepted and if $\Delta S \leq 0$, the non-optimal move is accepted with a probability *p*. In the course of optimization the effective temperature of the system goes down which results in lowering the probability *p* of accepting non-optimal moves:

$$p = \left[\frac{C - \Delta S}{\sqrt{m}} \right] \qquad (2)$$

where, $C = 25$ is a constant and m is the trial move count from the beginning of the optimization. If the move is accepted, the change in alignment becomes permanent and if not, the change is discarded and the method proceeds to the next trial move. Trial moves are attempted until convergence, that is, there is no further improvement in the score for a consecutive m_{conv} number of steps, or the trial move count reaches a maximum of m_{max} steps. Here, values of m_{conv} and m_{max} are based on the number of structures being aligned and the length of the seed alignment.

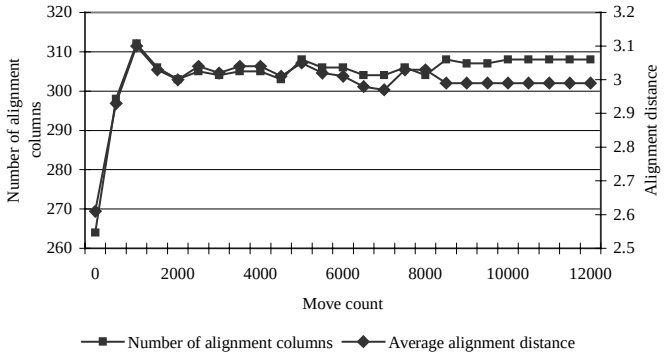

Figure 3. Changes in alignment distance and number of alignment columns during Monte Carlo optimization

Figures 3 & 4 depict a typical Monte Carlo optimization run for the protein kinase family (1CDK:A). In the first 1000 iterations there has been a very sharp rise both in alignment score (Fig. 4) and in the number of aligned columns (Fig. 3). Average alignment distance (Fig. 3) shows the same pattern as the number of alignment columns up to 5000 iterations (Fig. 4) and stabilizes at a lower level after

that. An increase in the distance upon an increase in the number of alignment columns is almost invariable among all families tested. This is because our seed alignment already has fragments aligned by the CE algorithm and the major goal of the current algorithm is to align regions not aligned by CE, regions that are inherently structurally distant. A detailed analysis of the relationship between the average alignment distance and the number of aligned columns is given in Figure 5.

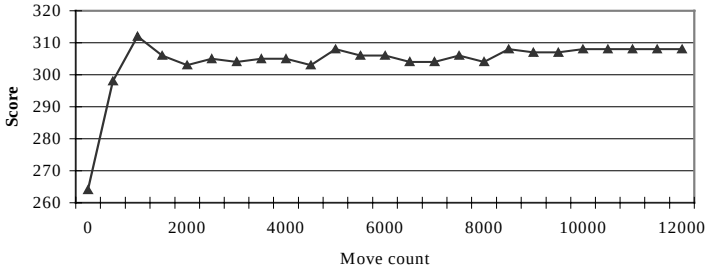

Figure 4. Pattern of the alignment score change during Monte Carlo simulation

Further analysis of the convergence by starting the optimization using a different seed number for the random generator showed that improvement in the alignment score, as well as changes in number of alignment columns and alignment distance, were consistent in multiple (five) runs.

Table 1. Effect of Monte Carlo optimization for 66 selected protein families.

Parameter	Before Optimization	After Optimization	% Change
Number of alignment columns	154.61	172.91	12 ↑
Total alignment length	426.63	333.84	22 ↓
Alignment score	509.7	859.13	69 ↑
Average alignment distance	2.63	3.09	17 ↑

3.2 Analysis of multiple protein families

The MC algorithm was run against 66 protein families selected as described above and the results are given in Table 1. The average increase in the alignment score is 69% which is accompanied by a 12% increase in the number of aligned columns and a 22% decrease in the total alignment length. As expected, the average alignment distance also increased by 17% with the increase in number of aligned columns. Increase in the average alignment distance is contrary to the goals of our

algorithm; however, a reasonable tradeoff with distance seems indispensable to accomplish the goal of this algorithm.

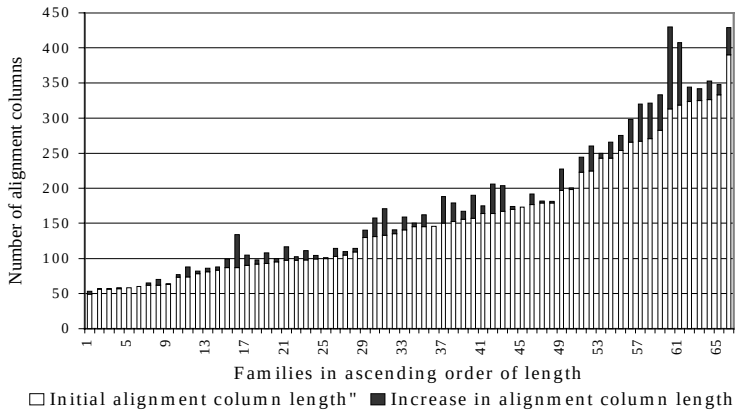

Figure 5. Improvement in the number of alignment colums with Monte Carlo optimization of 66 families in the PDB

Figure 5 shows the distribution of improvements in the number of columns for each of the 66 families under study, sorted in ascending order of alignment columns. The MC optimization provides greater improvements for the families with more alignment columns. This is explained by larger alignments having more variation and the algorithm exploring a wider searching space.

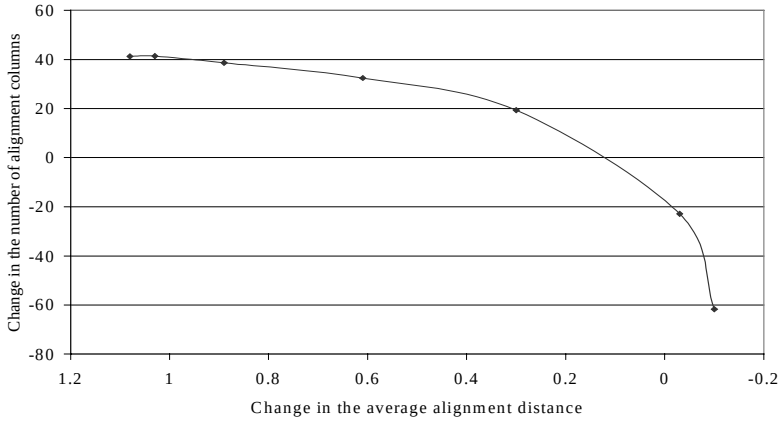

Figure 6. Relationship between number of alignment columns and average alignment distance in Monte Carlo optimization

To further study the relationship between the average alignment distance and the number of alignment columns, we ran the algorithm against the 9 families with 7 different settings of scoring function (d_0 distance in equation 1). Figure 6 plots the change in the number of columns (Δc) against the change in the average alignment distance (Δd). As seen from the previous results (Fig. 3), there is a direct relation between Δc and Δd, i.e., an increase in one always leads to an increase in the other and vice versa. The Δc decreases rapidly as Δd approaches zero because, for the majority of the families tested, the initial average distances are in the range of 1-3 A°. When $d_0 < 3$, the score for many alignment columns become negative and further optimization results in rapid loss of alignment columns from the blocks.

3.2.1 Performance of the algorithm

The algorithm was run on a single processor of SUN HPC6000 with 2 GB of main memory. The time taken (CPU time) for converging a protein family of 15 chains with an average residue length of 295, was 4.9 minutes. Computation time grows quadratic with the number of structures in the alignment. It also depends on a number of other parameters, among which the available search space (the ratio between aligned to the non-aligned region in the alignment) and the initial alignment distance are the key factors (Data not shown due to space constraints).

3.3 Analysis of specific protein families

To further evaluate the contribution of this MC algorithm to improvements in multiple structural alignments, two specific protein families have been selected for further study: protein kinases and aspartic proteinases. MC optimized alignments have been compared to assembled pairwise CE alignments and to curated (manually optimized) alignments. Curated alignments for aspartic proteinases have been obtained from HOMSTRAD[11]. Curated alignment for protein kinases have been built in a separate research effort, briefly described below.

3.3.1 Analysis of an alignment of protein kinases

An assessment of the MC algorithm was performed by aligning a set of 17 divergent protein kinase catalytic cores taken from the PDB. The representatives were chosen such that the sequence identity upon structural alignment with CE was <50% between any two structures. The protein kinases are composed of a small, mostly beta sheet N-terminal domain and a larger, mostly alpha-helical C-terminal domain joined by a flexible hinge region. Residues important for ATP binding and phosphotransfer line the active site cleft between the domains[12]. The MC algorithm has improved the alignment score by 153%, which is accompanied by a 17% increase in number of aligned columns, a 19% increase in average alignment distance, and a 29% reduction in the total alignment length.

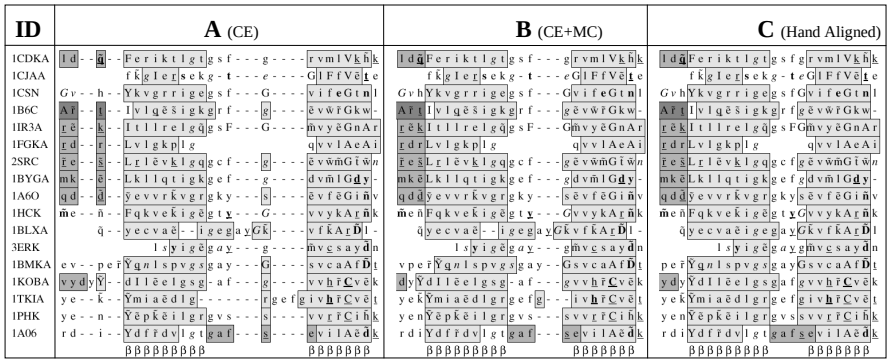

Fig. 7: Alignment of strand 1, the glycine rich loop, and strand 2 of the protein kinases, by standard CE, CE + MC, and hand curated alignment. Alignment is shown in the JOY format[13] which annotates the sequence alignment for structural features. Shaded boxes: *light gray:* β-strand, *medium gray:* 3-10 helix, *dark gray:* α-helix. Residue (letter) characteristics: *uppercase:* solvent inaccessible, *lowercase:* solvent accessible, *italic:* positive Φ, *breve(˘):* cis-peptide, *tilde(~):* hydrogen bond to other sidechain, *bold*: hydrogen bond to mainchain amide, *underline:* hydrogen bond to mainchain carbonyl. Blank regions signify portions of the sequence which lack atomic coordinates in the structure.

A hand-curated alignment was established by careful examination and adjustment of the automated CE structural alignments, curated alignments provided in the HOMSTRAD database[11] and two reviews[12,14]. Much of the MC alignment largely agreed with the curated alignments, particularly in the C-terminal subunit, a less flexible portion of the kinase structure. Two alignment sections of the N-terminal subunit are highlighted here (Fig. 7, Fig. 8). These sections represent *the most challenging sections* of the protein to align because of the positional variability seen in this region of the molecule[15]. They also illustrate the improvements the MC optimization provides over the standard CE protocol. Consider the examples:

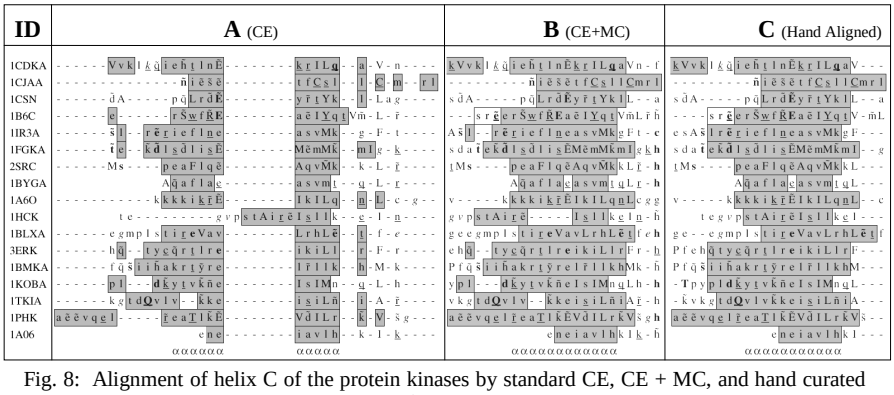

Fig. 8: Alignment of helix C of the protein kinases by standard CE, CE + MC, and hand curated alignment. Alignment is shown in the JOY format. Shaded boxes: *medium gray:* α-helix, *white:* 3-10 helix.

Example 1: Strand 1 and Strand 2 of the glycine-rich loop (Fig. 7). This region of the kinase domain is flexible and often in different conformations[15]. However, it contains the well-conserved GxGxxG motif, which is important for the binding of ATP in the active site[12]. With the exception of one structure (1CJA:A) it should be aligned without gaps to properly align this motif. Standard CE alignment splits off some of the sequence leading up to strand 1, and unnecessarily separates off a row of conserved glycines in the loop between strands 1 and 2 (Fig. 7A). The MC alignment compresses the sequence leading up to strand 1, and closes the gap which causes the glycine displacement in CE (Fig. 7B). However, MC does not correct the misaligned glycine residues seen in some structures in the original CE alignment. (Figs. 7A-C).

Example 2: Helix C (Fig. 8). This helix is found at different angles in the various protein kinase structures[15] making it difficult to align. However, it should be aligned without gaps, and a highly-conserved Glu residue should be lined up in all structures. The standard CE alignment produces multiple small gaps in the alignment at the ends of the helix, as well as one large gap based on 1HCK (Fig. 8A). Inspection of 1HCK reveals that helix C is displaced and rotated to a particularly large degree in this structure. The MC alignment compresses most of the gaps at the ends of the helix and realigns the improperly gapped section (Fig. 8B). However, it does not correct the misaligned Glu residues seen in some structures in the original CE alignment (Figs. 8A-C).

3.3.2 Analysis of an alignment of aspartic proteinases

We have selected aspartic proteinases, which are composed of a high proportion of beta sheets and relatively few alpha helices, as a second family for testing the MC algorithm. Important members of this family are renins (1BBS:_, 1SMR:A), pepsins (5PEP:_, 1PSN:_, 1JXR:A, 1MPP:_, 1AM5:_) and proteinases (3APP:_, 4APPE:_, 2APR:_, 2ASI:_) which are associated with several pathological conditions in humans. We have used the same 12 structures as classified by HOMSTRAD under this family for ease of comparison and reference. These structures have an average sequence similarity of 37%. Seed alignments were constructed from CE pair-wise data as explained previously, using 3APP:_ as the master.

The MC algorithm has improved the alignment score by 19%, which is accompanied by a 10% increase in number of aligned columns, a 13% increase in average alignment distance, and a 35% reduction in the total alignment length. Many improvements were observed in the overall alignment especially in the areas that CE failed to align properly. Due to space limitations, examples of only two major improvements are presented in figures 9 & 10 (For an explanation on the structural features of residues in JOY format refer to Figure 7). Consider the examples:

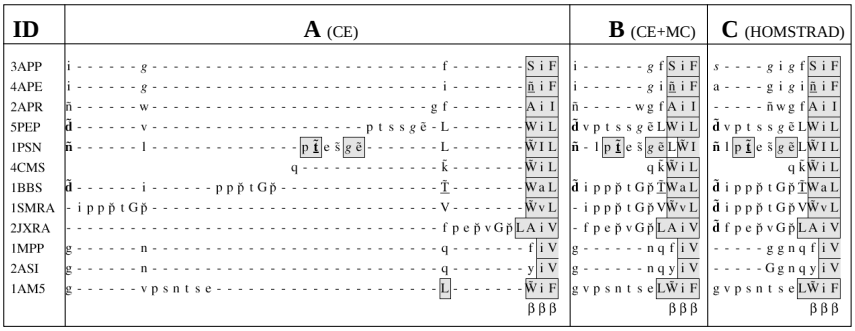

Figure 9. Comparison of multiple alignments generated by CE, MC and HOMSTRAD, in the poly-proline segment of the aspartic proteinases family. Shaded boxes: *light gray:* β-strand.

Example 1: Aspartic proteinases exhibit a bilobal structure with an active site cleft in the middle of two lobes. On the opposite side of the active site cleft, there is a "poly-proline" loop contributed by the C-terminal domain. In the case of renins, the sequence contains ~P-P-P-T-G-P~ (although the analogous structure in some aspartic proteinases contain fewer or even no proline residues) that influences binding of the S2' and S3' pockets in the active site cleft[16]. In the CE alignment this region is widely spread out with no alignment of the poly-proline residues (Fig. 9A). However, this region is well aligned by the MC algorithm in all the three chains (1BBS:_, 1SMR:A, 2JXRA) that contain this region (Fig. 9B) and these results compare well with the HOMSTRAD alignments (Fig. 9C).

Example 2: Another improvement is seen in the loop region between an α-helix and a β-sheet in the N-terminal lobe. As seen in figure 10, the CE alignment has spread out the residues that are well conserved in all but the first two structures (Fig. 10A), whereas the MC algorithm has realigned these residues (Fig, 10B) making it comparable to that of the HOMSTRAD alignment (Fig. 10C).

Summary

Multiple structure alignment is of increasing importance as we move into the era of structural genomics that will bring forth a large number of unannotated structures, for which automated functional assignments will be needed. This paper presents evidence that the MC algorithm proposed here can contribute in this regard. The application of MC improves some aspects of CE pair-wise alignments considerably, while other aspects are minimally affected. Most importantly, the MC optimization does not introduce any significant new errors into the alignment. The effects of the optimization are nearly always positive. The overall effect of the optimization is the compression of many of the gapped regions generated between and at the ends of secondary structural elements within the multiple alignments. The MC optimization has little effect within regions of the alignments where a small misalignment error is present well within a large block of aligned structure/sequence. These types of

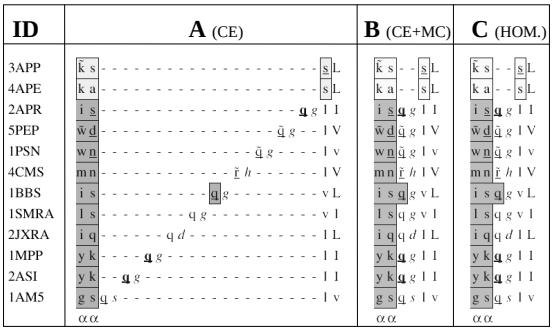

Figure 10. Comparison of multiple alignments generated by CE, MC and HOMSTRAD, in the N-terminal lobe of aspartic proteinases family. Shaded boxes: *medium gray:* α-helix, *white:* 3-10 helix.

errors do not seem to benefit from MC optimization. Current work seeks to: (i) further explore the behavior of MC by empirical means; (ii) compare MC to other automated multiple alignment techniques; (iii) provide a public database of aligned protein families.

Acknowledgments

This work was supported through NSF grant DBI 9808706.

References

1. W. Taylor and C. Orengo, *J. Mol. Biol.*, **208**, 1 (1989)
2. L. Holm and C. Sander, *J. Mol. Biol*., **233**, 123 (1993)
3. I.N. Shindyalov and P.E. Bourne, *Prot. Eng*., **11,** 739 (1998)
4. L.A. Mirny and E.I. Shakhnovich, *J. Mol. Biol*., **283**, 507 (1998)
5. N. Leibowitz et al, *ISMB*, 169 (1999)
6. D.F. Feng and R.F. Doolitte, *J.Mol.Evol.*, **25**, 351 (1987)
7. J.D. Thompson, F. Plewniak and O. Poch, *Nucleic Acids Res.*, **13**, 1682 (1999)
8. I.N. Shindyalov and P.E. Bourne, *CABIOS*, **13**, 487 (1997)
9. M. Gerstein and M. Levitt, *Protein Sci.*, **7**, 445 (1998)
10. I.N.Shindyalov and P.E. Bourne, *Proteins*, **38**, 247 (2000)
11. K. Mizuguchi et al, *Protein Sci*., **7**, 2469 (1998)
12. S.S. Taylor and E. Radzio-Andzelm, *Structure*, **2**, 345 (1994)
13. K. Mizuguchi et al, *Bioinformatics*, **14**, 617 (1998)
14. S.K. Hanks and T. Hunter, *FASEB J*., **9**, 576 (1995)
15. J.M. Sowadski et al, *Pharmacol Ther*., **82**, 157 (1999)
16. M.J. Humphreys and C. Berry, in *Structure and Function of Aspartic Proteinases: Retroviral and Cellular Enzymes*, Ed. James M.N.G. (PlenumPress, NewYork, 1998)