Frederick Lo
Biochemistry 218
Final Project Paper

Three Recently Developed Algorithms for Aligning Distantly Related Proteins

Introduction

The rate at which new protein sequences are discovered has long outpaced the rate at which those proteins are experimentally assigned functions. To speed the process of function assignment, protein sequences with unknown functions can be compared to sequences with known functions. If two protein sequences are shown to be highly related, then it follows that the protein functions may be closely related as well (Domingues et. al 2000).

Early sequence alignment algorithms such as Smith-Waterman and BLAST focused on comparing the residue sequences only in making alignments. However, function may be conserved even in sequences that appear to have diverged. Much recent effort has been devoted to developing algorithms that align distantly related proteins, also known as remote homologues. The three algorithms discussed in this paper consider three separate approaches to this alignment problem. The three algorithms are the Structure-Dependent Sequence Alignment, a hybrid Iterative-Parametric approach to suboptimal alignment, and the amino acid property-based Proximity Correlation Matrix alignment.

Structure-Dependent Sequence Alignment

Previous studies have shown that structural motifs in related proteins often remain highly conserved even in the presence of significant divergences in sequence (Jaroszewski et. al 2000). Therefore, integrating structural data generally leads to more accurate protein sequence alignments than using sequence data alone, especially for distantly related proteins. Structural alignment is frequently cited as the "gold standard" for sequence alignment (Sunyaev et. al 2004). Since protein structure and function are closely related, it is no surprise that a protein's amino acid sequence is much easier to

determine than the protein's structure, such that the number of known protein sequences far outstrips the number of known protein structures (Jaroszewski et. al 2000). Unless the structures of both the query and template proteins are known, structural alignments cannot be conducted. The Structure-Dependent Sequence Alignment, or SDSA, algorithm attempts to bridge this gap (Yang 2002).

SDSA is a sequence-structure alignment algorithm that is adapted from the Needleman-Wunsch global sequence alignment algorithm. The Needleman-Wunsch algorithm can be expressed with the following equation:

$$H_{ij} = D_{ij} + \max[H_{i+1,j+1}, \max_{k=i+2,Na}(H_{k,j+1} - g((i,j),(k,j+1))), \max_{k=j+2,Nb}(H_{i+1,k} - g((i,j),(i+1,k)))] \quad \text{(Eq. 1)}$$

D is the sequence-sequence amino acid substitution matrix, where D_{ij} is the score for substituting the template residue at position j with the query residue at position i. H is the scoring matrix, where H_{ij} is the maximum score for all sequences rooted at (i, j). Na and Nb are the number of residues found in the query and template sequences, respectively. The function g((i, j)(k, j+1)) is the penalty for inserting a gap in the query sequence, whereas g((i, j)(i+1, k)) is the penalty for deleting a subsequence in the template. The formulae for determining g in SDSA will be discussed later. The maximal value in H indicates the best alignment, and the entire alignment can be obtained by traversing H diagonally downward from left to right, where position (1, 1) represents the top left corner of H (Yang 2002).

At the core of SDSA is the construction of a structure-based amino acid substitution matrix D. Since the structure of the template sequence is known, residue j in the template sequence can be identified as belonging to an α-helix, β-strand, or coil region. Thus, the following conditional equations were developed for D:

$$D_{ij} = q_j \lfloor A_{ij} + fP_i(\alpha) \rfloor + w_j \quad \text{when residue j belongs to an α-helix} \quad \text{(Eq. 2a)}$$

or $$D_{ij} = q_j \lfloor B_{ij} + fP_i(\beta) \rfloor + w_j \quad \text{when residue j belongs to a β-strand} \quad \text{(Eq. 2b)}$$

or $$D_{ij} = C_{ij} + fP_i(c) \quad \text{when residue j belongs to a coil region.} \quad \text{(Eq. 2c)}$$

A, B, and *C* are amino acid substitution matrices for residues in α-helices, β-strands, and coil regions, respectively. Yang used a database of protein structure fragments to obtain

170,673 pairs of local structure alignments, which consisted of 503,466, 1,130,201, and 516,046 pairs of aligned α-helices, β-strands, and coil regions, respectively. By processing these pairs with the Protein Informatics System for Modeling (PrISM.1) structural alignment procedure also developed by Yang and utilizing formulas used by Henikoff and Henikoff (1992) to derive the BLOSUM substitution matrices, the A*, B,* and *C* matrices were constructed (Yang 2002).

The functions $P_i(x)$ in Equation 2, where x is α, β, or c, represent the log-odds probability that amino acid i will be found in an α-helix, β-strand, or coil region, respectively. These probabilities were derived from the pairings generated for the *A, B,* and *C* substitution matrices above. The general formula for $P_i(x)$ is:

$$P_i(x) = \ln\left(\frac{n_i(x)/n_i}{n(x)/n}\right)$$

where $n_i(x)$ is the number of amino acid type i found the sequences of structure type x, n_i is the number of amino acid type i found in the total set of residues from all pairings, n(x) is the number of residues found in all pairings of type x, and n is the total number of residues found in all pairings. The numerator $n_i(x)/ n_i$ can be interpreted as the probability that a given amino acid type i will be found in structure type x, while the denominator n(x)/n can be interpreted as the probability that a random amino acid will be found in structure type x (Yang 2002).

Parameters q, f, and w_j in Equation 2 are user specified. Parameter q is a measure of whether residue i is exposed to solvent or buried. The residue is considered buried if less than 20% of its surface area is exposed to solvent. The weighting parameter f for the log-odds probability should ideally have a value of 1 but is present to allow flexibility to improve alignment accuracy. Parameter f will be determined by using training sets. Parameter w_j provides extra weight for α-helix and β-strand residues, to differentiate them from coil regions. Although Yang claims that w_j can be varied given information about the template structure, no explanation was given on how this would be done. Parameter w_j is generally set to 1 (Yang 2002).

With the amino acid substitution matrix D now constructed using Equation 2, it can be placed in Equation 1. However, the gap insertion and deletion components of

Equation 1 also need to be adjusted to take structural considerations into account. For gap insertion:

$$g((i,j),(k,j+1) = -a - b(k-i-2) - c(j,j+1) \qquad \text{(Eq. 3)}$$

where the gap is inserted between positions i+1 and k-1 of the query sequence, and residues i and k in the query sequence are matched with residues j and j+1 in the template sequence. Variable a is the gap opening penalty and b is the gap extension penalty. Variable c is the penalty for opening a gap when j and j+1 are found inside an α-helix or β-strand such that c(j, j+1) = 100 for such a case and 0 otherwise. The penalty due to c prevents gaps from opening within α-helices or β-strands, which generally remain highly conserved between related proteins (Jaroszewski et. al 2000). The deletion equation takes on a different form:

$$g((i,j),(i+1,k) = -a - b(k-j-2) - d * DIST(j,k) \qquad \text{(Eq. 4)}$$

where the deletion occurs between residues j and k in the template sequence, and residues i and i+1 in the query sequence are matched with residues j and k in the template sequence. DIST(j, k) is the distance between alpha carbons of residues j and k, while d is an empirically determined parameter. The DIST penalty prevents large deletions from occurring in the template sequence (Yang 2002).

To test the alignment accuracy of SDSA, Yang first used the algorithm on a training set of 412 sequence pairs with known structures to determine optimal values for parameters a, b, d, f, and q_j, then evaluated the algorithm on a testing set of 1421 sequence pairs with known structures. The protein sequences in the training set were taken from the SCOP database and had to meet certain requirements. The pairs had to have less than 40% identity base on structural alignment and have a protein structural distance (PSD) of less than 1, and the shorter sequence of the pair had to be no less than 80% in length of the longer sequence. Also, the pairs had to contain at least two protein structural components (e.g. two α-helices, or an α-helix and a β-strand), and the sequence similarity had to be undetectable by PSI-BLAST with an E-value cutoff of 100. With these requirements, pairs in the training set ended up having an average identity of 8%. After rigorous testing on combinations of a small set of values for each parameter, the optimal values were determined to be q = 1.5, f = 0.6, d = 0.2, a = 0.4, and b = 0.1. Of

the five parameters, a, b, and d had the greatest effects on accuracy, while f had a smaller effect and q had little effect (Yang 2002).

To complete the evaluation of SDSA, Yang tested SDSA against PSI-BLAST by aligning the 1421 sequence pairs in the testing set using both algorithms. The testing set sequence pairs were taken from PDB and had to meet the same requirements as the training set. The average identity of the training set pairs was 13%, and 96% of all pairs had identities less than 30%. The testing set was divided into four groups based on their E-value from PSI-BLAST. For some E-value e, the groups were divided as follows: $e < 10^{-6}$, $10^{-6} < e < 10^{-2}$, $10^{-2} < e < 10^{-1}$, and $e > 10^{-1}$. The average pairwise identity of the groups were 24%, 18%, 12%, and 7%, respectively. Using the PrISM.1 structural alignment of sequence pairs as the gold standard, alignment quality was measured by S_0, the ratio of correctly aligned query residues to the total number of query residues (Yang 2002).

The evaluation showed that while PSI-BLAST outperformed SDSA in the $e < 10^{-6}$ group, SDSA was considerably more accurate for all other groups. The average S_0 values for PSI-BLAST for the groups, starting with the group having $e < 10^{-6}$, were 80.4%, 49.4%, 23.2%, and 8.0%. For SDSA, the average S_0 values were 77.3%, 55.0%, 38.1%, and 26.9%. While the S_0 percentages for SDSA are relatively low for the latter groups, they still represent a large improvement over alignment using PSI-BLAST alone (Yang 2002).

Rather than attempt to determine the E-value at which a user should switch between using PSI-BLAST and SDSA, Yang developed a simple hybrid approach that uses both algorithms and delivers performance that is better than that of using either algorithm by itself. This hybrid approach is a global structure-based sequence alignment algorithm. First, the query and template sequences are aligned using iterative PSI-BLAST. The resulting alignment is then translated into a 2D matrix V, where $V_{ij} = 1$ if query residue i is aligned to template residue j, and 0 otherwise. The modified amino acid substitution matrix D' is then constructed as follows:

$$D'_{ij} = D_{ij} + V_{ij} \quad \text{(Eq. 5)}$$

where i is the query residue at position i, j is the template residue at position j, and D is the structure-based substitution matrix constructed using Equation 2. D' is then used in

place of D in Equation 1. Running this hybrid approach on the testing set returned average S_0 values of 83.6%, 61.1%, 42.2%, and 27.4% for the four groups, all of which are higher than the percentages reported above for PSI-BLAST and SDSA run alone. SDSA is thus a valid extension to PSI-BLAST (Yang 2002).

Hybrid Iterative-Parametric Suboptimal Alignment

It may be possible to enumerate and evaluate all possible sequence alignments if the query and template sequences are both short. However, as the sequences lengthen, it quickly becomes unfeasible due to limitations in both time and space to examine the entire search space of possible alignments. For near optimal sequence alignment, being able to effectively limit the search space to likely alignment candidates greatly reduces the amount of computational time and space required to find an acceptable suboptimal alignment by focusing the search. The goal of a search space limiting algorithm is to generate a set of suboptimal alignments that contains one or more alignments that is "better" than the optimal alignment generated by a sequence alignment algorithm, a situation that can often occur for distantly related proteins. Jaroszewski, Li, and Godzik present two general methods for generating sets of suboptimal alignments, then demonstrate that a hybrid of the general methods returns better sets of suboptimal alignments.

The immediately apparent problem is that a set of alignments is useless unless the "best" alignment in the set can be identified. One possible approach to this identification is to build protein models for all the alignments in the sets and then to determine and compare the self-threading energy of each model. An implementation of this methodology by the Jaroszewski group called the Multiple Model Approach was used to evaluate the performance of the search space limiting algorithms discussed in this section (Jaroszewski et. al 1998).

The first of the two general near-optimal sequence generating methods is called iterative elimination. It modifies alignment algorithms that generate sequence-sequence amino acid substitution matrices, such as such as Needleman-Wunsch, for suboptimal alignment. Jaroszewski uses the term "similarity matrix" rather than amino-acid substitution matrix. The two matrix types appear to be analogous, although there are

some differences. The sequence-sequence similarity matrix can apparently be constructed from a substitution matrix and dynamic programming. Like an amino-acid substitution matrix, the similarity matrix measures similarity between pairs of query and template residues. However, while a high value in the substitution matrix indicates a valid substitution, a low value in the similarity matrix apparently indicates high similarity between residues. The iterative elimination algorithm begins by building a similarity matrix with the selected alignment algorithm. The best alignment (lowest sum) is determined by traversing the similarity scores in Needleman-Wunsch fashion, and this becomes the first alignment in the set of found alignments. The iterative algorithm continues by adding a value Δ to all cells in the similarity matrix visited by the first found alignment. The new best alignment is determined and added to the set of found alignments, and a slightly smaller Δ value is added to all cells visited by the second found alignment. Repeating this procedure yields additional suboptimal alignments. Gradually reducing Δ prevents suboptimal alignments from diverging too far from the initial "optimal" alignment, where a good starting Δ value is ¼ the average absolute value of the similarity matrix values (Jaroszewski et. al 2002).

The second of the general sequence generating methods is called the parametric approach. The approach is based on observations that changing alignment parameters, such as gap opening and extension penalties or weighting constants used to calculate the similarity (or substitution) matrices, will alter the resulting alignment of remote homologues. The method consists of aligning protein sequence pairs multiple times using different alignment parameters. The rationale is that there is no single set of parameters that will return the best alignment for every combination of protein sequences. Therefore, by using multiple sets of parameters rather than a single set, the parametric approach will generate sets of alignments that have a higher probability of containing at least one improved alignment. The parametric approach begins with the initial generation of an alignment and attached similarity matrix using some alignment algorithm. The parameters are then used to manipulate the similarity matrix, and alignments are obtained after each manipulation (Jaroszewski et. al 2002).

The hybrid approach proposed by Jaroszewski combines the iterative and parametric approaches. First, the parametric approach is applied. Sets of suboptimal

alignments are generated by aligning a given protein sequence pair using a single combination of non-gap-penalty-related parameters and many combinations of gap-penalty-related parameters. Since small adjustments to the gap parameters produce large changes in alignment, each set contains highly varied alignments (Shibuya and Imai 1997). The best alignment and accompanying similarity matrix is determined from each set, and further alignments are obtained from the similarity matrices by applying the iterative elimination method.

To test the effectiveness of the three approaches above, the authors isolated 742 protein pairs from the SCOP database. The pairs had ~45% identity or less. All three methods require an initial alignment and similarity matrix, so the profile-profile alignment algorithm Fold and Function Assignment System (FFAS) developed by the Jaroszewski group was utilized (Rychlewski et al. 2000). Upon alignment of a protein sequence pair, FFAS returns a z-score, which can be related to the global sequence similarity of the protein pair. According to the authors, a z-score of 14 or higher indicates a high degree of similarity, such that suboptimal alignments would not show improvement over the initial optimal alignment. Scores between 14 and 7 indicate a moderate degree of similarity, although the alignment accuracy may be low. Scores between 7 and 2 correspond to low similarity pairs which require additional information for accurate alignments. Finally, Jaroszewski asserts that scores below 2 cannot be aligned with currently available methods. The 742 SCOP protein pairs were divided into four groups based on each pair's FFAS z-score. The authors determined that the moderately similar protein pairs with z-scores between 14 and 7 would stand to gain the most improvement through use of the suboptimal alignment methods, and the test results for these pairs only were reported.

The testing procedure for the iterative elimination approach was exactly as described before, using FFAS to perform the initial alignment and then penalizing the similarity matrix using the last found alignment during each iteration. 1000 iterations were performed on each sequence pair as Δ was reduced from ¼ the average absolute value of the similarity matrix values to 0.01. The testing procedure for parametric approach also began with an FFAS alignment. The similarity matrix was calculated with the following formula:

$$Sim_{ij} = P_{ij} + \beta B_{ij} + \lambda L_{ij} \quad \text{(Eq. 6)}$$

where Sim_{ij} is the similarity value between the query residue at position i and the template residue at position j, P_{ij} is the FFAS profile-profile matching term for the two residues, B_{ij} is the "burial term" for the two residues taken from a Jaroszewski threading algorithm, and L_{ij} is the "local propensity term" for the two residues taken from a Jaroszewski threading algorithm. β and λ are weighting parameters. Each of the four parameters (β, λ, and the gap opening and extension penalty parameters) could take on one of four values, such that each sequence pair was processed 256 times using the parametric approach. The exact nature of a "burial term" or a "local propensity term" is not as important as the demonstration that varying the parameters will alter the score for a given alignment and thus change the best (lowest scoring) alignment for different sets of parameter values. Most likely any parameter-using procedure that generates similarity or substitution matrices, such as Yang's D matrix generating equations from Equation 2, can be adapted for use in the parametric approach. As for the hybrid approach, the alignments generated by the parametric approach were divided into groups, and the best alignment from each group was processed using the iterative elimination method to generate further suboptimal alignments. A single sequence pair would generate approximately 4000 suboptimal alignments in all (Jaroszewski et. al 2002).

Within each approach, repeats of previously obtained alignments were discarded. As a result, the average number of unique alignments obtained for the iterative, parametric, and hybrid approaches were 275, 499, and 733, respectively. Of these unique alignments, the percentage that showed "significant improvement" when compared to the initial optimal alignment were 35%, 34%, and 48%. Significant improvement is defined as a 25% or greater reduction in root mean square distance (RMSD) when compared to the initial alignment and equal or greater contact map overlap (CMO), or a 25% or greater CMO increase accompanied by equal or less RMSD. In summary, each alignment of moderately similar sequence pairs generated by the iterative elimination or parametric approaches was significantly improved about 1/3 of the time, while the hybrid approach produced significantly improved alignments about ½ of the time. Thus, of the three approaches, the hybrid approach most effectively reduces the alignment search space (Jaroszewski et. al 2002).

Proximity Correlation Matrix Alignment

Structural considerations play an important roles in both SDSA and the Iterative-Parametric algorithms. In SDSA, structural patterns are integrated into the amino acid substitution matrix, while protein modeling is required to identify the best alignment in the set of alignments isolated by the Iterative-Parametric algorithm. Structure, however, is not the only factor that can be used to find good suboptimal alignments. The Proximity Correlation Matrix (PCM) algorithm designed by Grigoriev and Kim uses amino acid properties along with real or predicted secondary structures to perform the sequence alignment and identify distantly related proteins.

In designing PCM, Grigoriev and Kim made two simplifying assumptions. First, they assumed that local interactions around a particular residue in a given sequence strongly correlate with the local interactions for the corresponding residue in a remote homolog. Second, they assumed the conservation of the sequential arrangement of physical properties around a particular residue in a given protein sequence with the arrangement around the corresponding residue in a remote homolog. That is, the physical properties within a small window of residues surrounding corresponding amino acids in the two homologues should be highly similar. Five physical properties were considered: hydrophobicity, volume, normalized frequencies of α-helix, normalized frequencies of β-sheet, and relative frequency of occurrence. Of these five, using hydrophobicity returned the best results (Grigoriev and Kim 1999).

The assumptions made above allowed the proximity correlation matrix to be constructed. The correlation score corr for a physical property p was calculated over a window of size L for matched query residue i and template residue j, where i and j are at the center of their respective windows and L = 2*l* +1, with *l* being the number of amino acids sequentially ahead and behind the central residue to be included in the window. Then:

$$corr(i,j) = \frac{1}{2l+1} \frac{\sum_{m=-l}^{l} (p_{i+m} - \overline{p}^{i})(p_{j+m} - \overline{p}^{j})}{\sigma^{i}\sigma^{j}} \qquad \text{(Eq. 7)}$$

where $\overline{p}^{i}$ and $\overline{p}^{j}$ are the average property values in the residue i and j windows, respectively, and σ^{i} and σ^{j} are the property standard deviations for those windows,

respectively. To reduce the probability that high correlations will appear by chance, correlation values are meaningful only when residues i and j belong to the same type of secondary structure. When real secondary structures are not available, they should be approximated using a secondary structure prediction algorithm (Heringa 2000). The correlation score for (i, j) is set to zero otherwise. The overall correlation score for an entire alignment is calculated by using the classic Needleman-Wunsch algorithm on the correlation matrix. Next, the overall correlation score S_{qt} between a query sequence q and target sequence t is converted into a Z-score as follows:

$$Z_{qt} = (S_{qt} - \overline{S}_q) / \sigma_s \qquad \text{(Eq. 8)}$$

where $\overline{S}_q$ and σ_s are the average score and standard deviation for alignments between the query and all targets. To determine remote homologues for a given query sequence, the Z scores for alignments between the query and all targets are arranged by increasing Z scores and the following heuristic is used:

$$Z_{cutoff} = Z_i \text{ if } Z_i - Z_{i+1} > \varepsilon \text{ and } Z_j - Z_{j+1} \leq \varepsilon$$
$$\text{for any } j > i \text{ and } Z_j > 0 \text{ and } \varepsilon \text{ is some constant} \qquad \text{(Eq. 9)}$$

where the target sequences in alignments having Z-scores greater than Z_{cutoff} are considered remote homologues of the query. When ordering Z-scores for alignments involving a particular query sequence, the Z-scores tend to clump, with large gaps appearing between clumps. The heuristic identifies the first gap that is at least ε units and delimits that gap as the separator between remote homologues and non-remote homologues (Grigoriev and Kim 1999).

To test the PCM algorithm, Grigoriev and Kim selected representative (query) proteins from 64 fold families found in the SCOP database. 1390 protein sequences were obtained from the SCOP and FSSP databases, where the identity between any two of these sequences was 25% or less. At least three remote homologues for each of the 64 representative proteins were present in the 1390 protein testing set, and 420 remote homologues were present in all. Both real and predicted secondary structures were used to generate the correlation matrices during testing. With ε set to 0.9, the number of true remote homologues found with PCM using real and predicted secondary structures were 178 and 167, respectively. PCM was able to identify more than 40% of remote

homologues in both cases. PCM was also tested against PSI-BLAST using the 64 representative proteins and 420 remote homologues as queries. PSI-BLAST did a better job identifying remote homologues when sequence identities were greater than 15%, while PCM was more accurate when sequence identities dropped below 15%. PCM is thus another possible extension to PSI-BLAST (Grigoriev and Kim 1999).

Analysis and Conclusions

Of the three algorithms, which is best? Space and time considerations do not play a part, as they are equal for all three algorithms. Since the most expensive step in each algorithm appears to be the construction of the MxN substitution/similarity/proximity matrix, where M and N are the lengths the sequences being aligned, the space cost is O(MN). The construction of each cell in the matrix requires constant-time lookup in some other matrix or database, so the running time is also O(MN). In actuality, each algorithm has its own strengths and weaknesses, so deciding on which algorithm to use differs depending on the situation.

SDSA is able to at least partially align protein pairs that have very low sequence identity. For example, it was able to correctly align an average of 26.9% of residues in sequences that had an average identity of 7%, and the percentage of correctly aligned residues rapidly increased as sequence identity increased. SDSA is thus able to detect very weak relationships between distantly related proteins. However, SDSA also requires that the template protein structure be known. If the template structure is unknown and cannot be accurately predicted, then SDSA cannot be used to perform the sequence alignment. The applicability of SDSA is therefore limited by the availability of structural information.

The Iterative-Parametric suboptimal alignment approach requires that protein sequences have moderate sequence identity in order to limit the search space effectively, as concluded by the algorithm's authors themselves. The approach is not useful for directly aligning very distantly related proteins, because the approach's success is highly dependent on an accurate initial alignment, a requirement that can be difficult to achieve for very remote homologues. On the other hand, since the Iterative-Parametric approach does not require any secondary structure information on either the query or template, the

approach can be applied to any sequence pair (with the caveat that not all returned alignments may be accurate). Consequently, the Iterative-Parametric approach has a lower accuracy but larger domain than SDSA. The ideal situation to use this approach is when the sequence pair has moderate identity and no structural information is known for the pair. Moreover, since the main factor limiting the effectiveness of this approach is the need for a decent initial alignment, the approach seems to be a good choice for refining a fairly accurate alignment generated by a separate algorithm. For example, SDSA could be used to perform the initial alignment on a set of very distantly related proteins. This initial alignment could then be further processed and refined by using the Iterative-Parametric approach.

PCM is also able to identify very distantly related proteins, as it was able to perform better alignments than PSI-BLAST when sequence identity dropped below 15%. Unfortunately, the domain of PCMs is even more restricted than that of SDSA. First, PCM requires secondary structures for both sequences in a sequence pair. Second, PCM can only identify homologues for a query sequence when the query is aligned with a group of target sequences. In other words, sequence alignment quality for a single alignment between a query and target cannot be determined, as Z_{qt} and Z_{cutoff} (Eqs. 8 and 9) have no meaning for the single alignment. To determine whether or not an alignment between a query and target is a good alignment, PCM must compare that alignment to all other alignments involving the query sequence. Furthermore, in edge cases where the target sequences are all remote homologues or all not remote homologues, PCM may return a considerable number of false positives or negatives. Application of PCM is thus limited to alignments where the secondary structures of both query and targets are known or can be predicted, where there are multiple targets to be aligned, and where the target sequences consist of a mix of remote homologues and non-remote homologues. PCM alignments could probably also be refined using the Iterative-Parametric approach.

Without the ability to perform straightforward sequence-sequence alignments to align distantly related proteins, researchers have had to exercise their creativity and biological know-how to the fullest. The three algorithms presented in this paper provided three different avenues to tackle this alignment problem. That each algorithm operates optimally only under specific conditions underscores the complexity and difficulty of the

remote homolog alignment problem and guarantees that it will be the focus of considerable future research.

Works Cited

Domingues, F.S., P. Lackner, A. Andreeva and M.J. Sippl. "Structure-based Evaluation of Sequence Comparison and Fold Recognition Alignment Accuracy." Journal of Molecular Biology 297(4) (Apr 2000): 1003-1013.

Grigoriev, Igor V. and Sung-Hou Kim. "Detection of Protein Fold Similarity Based on Correlation of Amino Acid Properties." Proceedings of the National Academy of Sciences of the United States of America 96(25): (Dec 7 1999): 14318-14323.

Henikoff, S. and J.G. Henikoff. "Amino Acid Substitution Matrices from Protein Blocks." Proceedings of the National Academy of Sciences of the United States of America 89 (1992): 10915-10919.

Heringa, J. "Computational Methods for Protein Secondary Structure Prediction Using Multiple Sequence Alignment." Current Protein and Peptide Science 1(3) (Nov 2000): 273-301.

Jaroszewski, L, K. Pawlowski, B. Zhang and A. Godzik. "Multiple Model Approach: Exploring the Limits of Comparative Modeling." Journal of Molecular Modeling 4(10) (Oct 1998): 294-309.

Jaroszewski, Lukasz, Weizhong Li, and Adam Godzik. "In Search for More Accurate Alignments in the Twilight Zone." Protein Science 11 (2002): 1702-1713.

Rychlewski, L., L. Jaroszewski, L. Weizhong, and A. Godzik. "Comparison of Sequence Profiles. Structural Predictions with No Structure Information". Protein Science 8 (2000): 232–241.

Shibuya, T. and H. Imai. "New Flexible Approaches for Multiple Sequence Alignment." Journal of Computational Biology 4(3) (Fall 1997): 385-413.

Sunyaev, Shamil R., Gennady A. Bogopolsky, Natalia V. Oleynikova, Peter K. Vlasov, Alexei V. Finkelstein, and Mikhail A. Roytberg. "From Analysis of Protein Structural Alignments Toward a Novel Approach to Align Protein Sequences." Proteins: Structure Function and Bioinformatics 54 (2004): 569-582.

Yang, An-Suei. "Structure-dependent Sequence Alignment for Remotely Related Proteins." Bioinformatics 18(12) (2002): 1658-1665.