Marina Sirota

PROTEIN MULTIPLE ALIGNMENT

MOTIVATION:

To study evolution on the genetic level across a wide range of organisms, biologists need accurate tools for multiple sequence alignment of protein families. Given a set of biological sequences, a multiple alignment provides biologists with a way of identifying and visualizing patterns of sequence conservation, advancing both evolutionary and phylogenetic studies. Protein alignments in particular have been very useful in predicting protein structure and characterization of protein families. Obtaining accurate multiple alignments, however, is a very difficult computational problem because of the high computational cost and difficulty of evaluation of alignment quality.

BACKGROUND:

There are many types of sequence alignments. The two main branches of sequence comparison are global and local alignment. While global alignment examines the similarity between two sequences as a whole, local alignment looks at shorter highly conserved regions between two sequences. In this paper, I will concentrate on examining global alignments. Global alignment can be either pairwise, comparing two sequences, or multiple, comparing three or more sequences. Several differences between pairwise and multiple alignment problems should be noted. Pairwise alignment scores are evaluated by addition of match or mismatch scores for aligned pairs and affine gap penalties for unaligned pairs of amino acids or nucleotides. Affine penalties mean that the penalty given for a gap is not constant, but depends on its length. In contrast, there is no proper objective function for measuring alignment quality for protein multiple sequence alignment, making the problem more difficult. Pairwise alignment is easily optimized using dynamic programming; however, this is not practical in multiple alignments.

The fact that DNA codes for proteins should in theory imply that DNA and protein sequence alignments are very similar; however, we should note several differences between protein and DNA alignment. DNA sequences consist of only four different nucleotides, represented by four letters (A, G, C and T). Protein sequences contain 20 characters, or 20 different amino acids. For DNA regions to show good conservation, they need to be more than 50% similar, since the DNA alphabet is so small. For proteins, since the amino acid alphabet is much greater, only 20% similarity is sufficient to show good conservation. Protein and DNA aligners have very different tool specifications. For instance, protein aligners need to handle huge numbers of sequences (2-1000 sequences), while DNA aligners do not. However, the sequences that need to be examined by a protein aligner are much shorter (300-500 amino acids) compared to DNA sequences. When aligning proteins, we can use known structural alignments to validate our sequence alignments, where this is absolutely impossible when aligning DNA. However we want to compare sequences with low sequence identity which is much more difficult to do.

THE PROBLEM:

The problem of protein multiple alignment is to find the optimal pairing of letters between the given N sequences with a given scoring scheme for evaluating matching letters. More formally, given N sequences x^1, x^2, …, x^N, insert gaps in each sequence x^i, such that all sequences have the same length L and the score of the global map is maximum. This can be done using dynamic programming with time and space complexity of $O(2^N L^N)$, where L is the length of the sequence, which is not practical at all. Therefore more efficient algorithms are needed.

METHODS:

Here is a standard multiple alignment procedure that is used by many aligners:

1. Compute a distance matrix
2. Compute a guide tree
3. Get the sum-of-pairs using the substitution matrix
4. Use progressive alignment
5. Fix errors with iterative refinement

In this paper I will describe and compare several algorithms for multiple sequence protein alignment. I will talk about how these tools added to the basic algorithm above or how they differ from it and each other. I will start by describing three current standard methods, SAM, ClustalW and T-Coffee the first of which is a purely algorithmic technique, while the other two heavily rely on heuristics. I will then describe two new approaches MUSCLE and PROBCONS and compare them against each other as well as the three current standard methods mentioned above.

SAM:

The Sequence Alignment and Modeling system (SAM) is a collection of software tools for creating, refining, and using linear hidden Markov models for biological sequence analysis. The model states can be viewed as representing the sequence of columns in a multiple sequence alignment, with provisions for arbitrary position-dependent insertions and deletions in each sequence. The models are trained on a family of protein sequences using an expectation-maximization algorithm and a variety of algorithmic heuristics. A trained model can then be used to both generate multiple alignments and search databases for new members of the family.

As mentioned above, linear hidden Markov model is a sequence of nodes, each corresponding to a column in a multiple alignment. Each node has a match state, an insert state and a delete state (Fig. 1). A series of these states are used by each sequence to traverse the model from start to end. Being in a match state indicates that the sequence has a character in that column, while being delete state indicates that the sequence does not. Insert states allow sequences to have additional characters between columns.

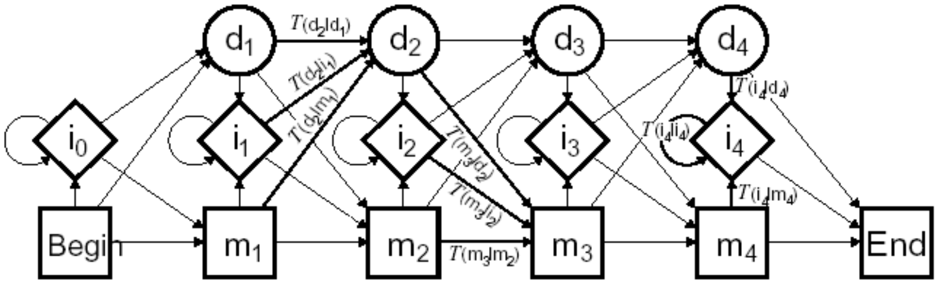


Fig.1 A Linear hidden Markov model.

The primary advantage of these models over other methods of sequence search is their ability to characterize an entire family of sequences. Just like the transition between states, each position has a distribution of bases. That is, these linear HMMs have position-dependent character distributions and position-dependent insertion and deletion gap penalties. The alignment of each of a family to a trained model automatically yields a multiple alignment among those sequences.

ClustalW:

ClustalW is one of the most widely used protein multiple aligner available today. First developed in 1988, it is one of the oldest multiple sequence tools.

The basic multiple alignment algorithm consists of three main stages:

1. All pairs of sequences are aligned separately in order to calculate a distance matrix giving the divergence of each pair of sequences
2. A guide tree is calculated from the distance matrix
3. The sequences are progressively aligned according to the branching order in the guide tree

ClustalW has greatly improved the sensitivity of the commonly used progressive multiple sequence alignment method, especially for the alignment of divergent protein sequences. ClustalW is the first tool that incorporates the idea of a weighted sum-of-pairs. When sequences that are to be compared are equally divergent from each other, it is fine to use an unweighted sum-of-pairs matrix; however, when some of the sequences being aligned are very similar, the alignment that is produced often has a bias. Thus individual weights are assigned to each sequence in a partial alignment according to the length of the branch in the tree. This is done in order to downweight near-duplicate sequences and upweight the most divergent ones. This technique lessens the contribution of redundant sequences in the alignment, getting rid of the bias.

ClustalW uses different amino acid substitution matrices at different alignment stages according to the divergence of the sequences to be aligned. This improves the alignment quality by picking different BLOSUM matrices to score the alignment according to similarity of the sequences.

Finally, ClustalW uses context-specific gap penalties in order to incorporate biological information into the alignment. For instance, residue specific gap penalties and locally reduced gap penalties in hydrophilic regions encourage new gaps in potential loop regions rather than regular secondary structure. Positions in early alignments where gaps have been opened receive locally reduced gap penalties to encourage the opening up of new gaps at these positions.

T-Coffee:

T-Coffee is the first method to substantially improve multiple alignment results on ClustalW and became the standard in alignment accuracy since 2000. This method is also based on the popular progressive approach to multiple sequence alignment. The most commonly used heuristic methods are based on the progressive-alignment strategy. The idea is to take an initial, approximate, phylogenetic tree between the sequences and to gradually build up the alignment, following the order in the tree. Although successful in a wide variety of cases, this method suffers from its greediness. Errors made in the first alignments cannot be recovered later as the rest of the sequences are added in. T-Coffee tries to avoid the most serious pitfalls caused by the greedy nature of the algorithm.

Several new important ideas were incorporated into this method. T-Coffee scores substitution matrices using an alignment library. T-Coffee pre-processes the data set of all pair-wise alignments between the sequences. This provides a library of alignment information that can be used to guide the progressive alignment. Intermediate alignments are then based not only on the sequences to be aligned next but also on how all of the sequences align with each other.

T-Coffee provides a simple and flexible means of generating multiple alignments, using heterogeneous data sources. The data from these sources are provided to T-Coffee through a library of pair-wise alignments. An example of such heterogeneity is data from local and global alignments (Fig. 2). The second main feature of T-Coffee is the optimization method, which is used to find the multiple alignment that best fits the pair-wise alignments in the input library.

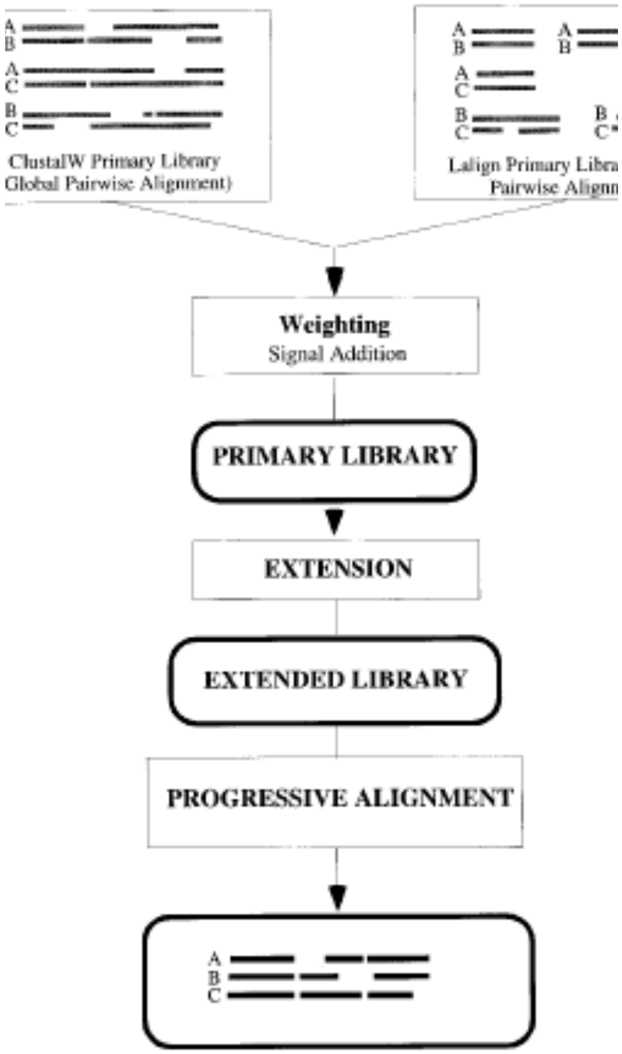


Fig. 2. **T-Coffee Algorithm**. Layout of the T-Coffee strategy; the main steps required to compute a multiple sequence alignment using the T-Coffee method. Square blocks designate procedures while rounded blocks indicate structures. (Notredame et al, 2000)

The first step in the T-Coffee process is the gathering of the pairwise alignments. A collection of such alignments is called a library. Each alignment is weighted by percent identity. Once the libraries have been computed, they can be pooled, extended and used to compute a multiple sequence alignment. During progressive alignment score for the alignment x_i to y_j is the sum of weights of alignments in library containing the alignment x_i to y_j.

Finally T-Coffee introduces the idea of consistency as one of its library extensions. This trick helps prevent errors in progressive alignment. The basic idea of consistency is to incorporate other pairwise alignment preferences during progressive alignment. Consider a region x_i which aligns well locally to both y_j and $y_{j'}$. Which one of these is the true alignment. Consider another sequence z, in which z_k aligns well to x_i. Aligning z with y to see whether z_k better aligns to y_j or $y_{j'}$ will give us more information about the correct alignment between x and y (Fig. 3).

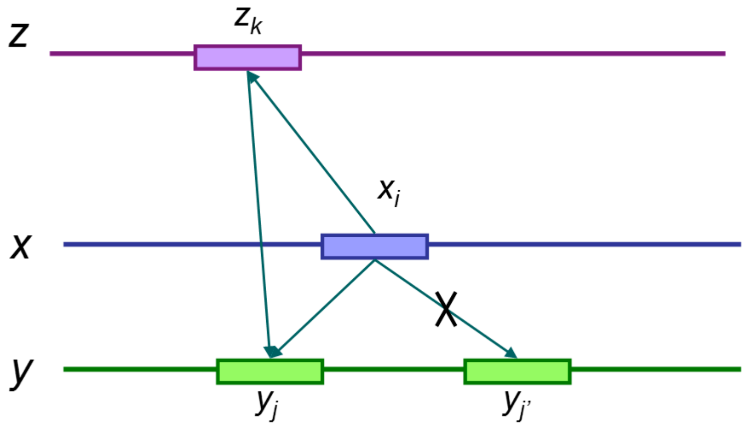


Fig 3. Consistency. (Batzoglou, 2005)

MUSCLE:

MUSCLE is a new very fast protein multiple sequence aligner which retains approximately T-Coffee accuracy. It uses a weighted log-expectation scoring for profiles and uses many optimizations to improve its performance.

First a progressive alignment is built, to which horizontal refinement is applied. There are three stages of the algorithm, draft progressive, improved progressive, and refinement. At the completion of each stage, a multiple alignment is available and the algorithm can be terminated. Implementing all three, however, is what allows MUSCLE to achieve significant improvement in accuracy and speed over previous techniques.

During the first stage, a progressive alignment of the sequences is built. Similarity of each pair of sequences is computed using k-mer counting. A k-mer is simply a continuous sequence of k letters. K-mer counting is recording the number of times identical k-mers appear in a sequence. A global alignment is constructed using the k-mers. The fractional identity of each of the sequences is determined. A binary tree is constructed where each leaf node is assigned a sequence and each internal node represents an alignment (Fig.4). The two child nodes are always aligned using profile-profile alignment (Fig. 5).

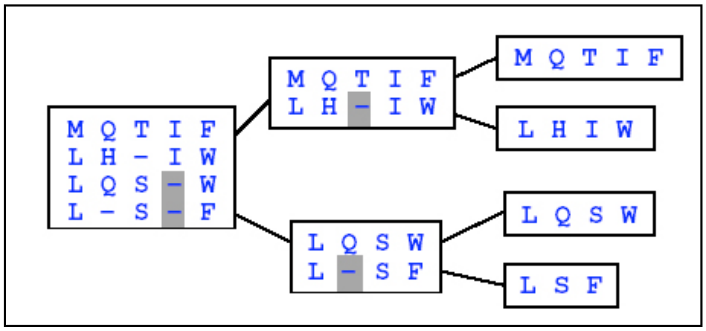


Fig. 4. Progressive alignment. (Edgar, 2004)

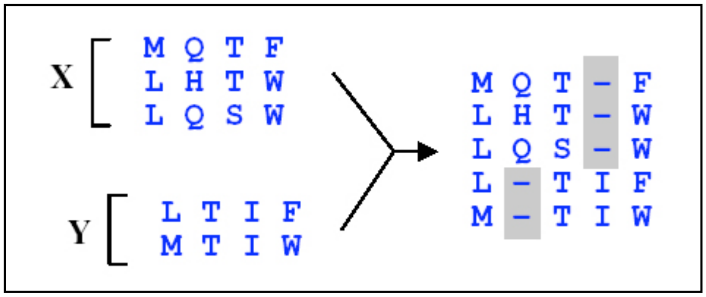


Fig. 5. Profile-profile alignment. Two profiles X and Y are aligned to each other so that the columns in both are preserved in the resulting alignment. Columns of gaps are inserted in order to achieve this result. The columns are scores using a function which assigns a high score to pairs of columns containing similar amino acids. (Edgar, 2004)

The second step of the algorithm is to improve the alignment that has been made. In order to do that, first the similarity of each pair of sequences is computed using fractional identity from the mutual alignment. A different tree is constructed by applying a clustering method to the distance matrix. The two trees are then compared and a set of nodes for which the branching order has changed is identified. If the order has not changed, then the original alignment is retained. Otherwise a new, more accurate alignment is built.

The t final step of the algorithm is iterative refinement. An edge is deleted from a tree, dividing the sequence into two disjoint subsets (Fig. 6). The profile (multiple alignment) of each subset is extracted and the profiles are then re-aligned to each other using the method described above. If the score has increased, the alignment is retained. Otherwise, it is discarded. The algorithm terminates at convergence.

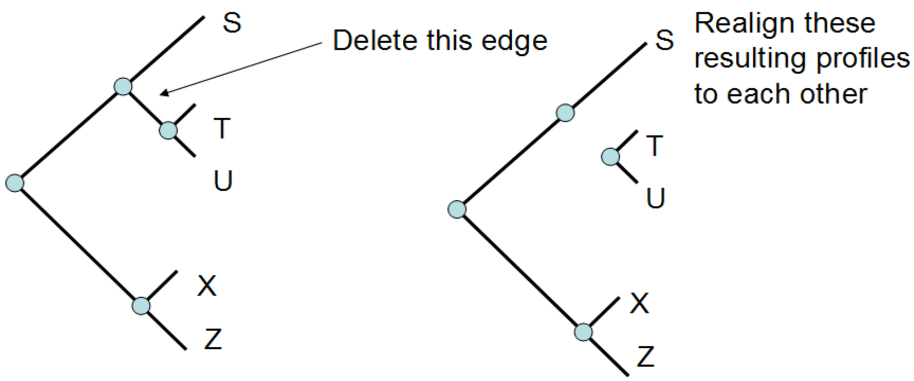


Fig. 6. Iterative Refinement

MUSCLE shows significant improvement over previous algorithms. It has O(N^2 + L^2) space and O(N^4 + NL^2) time complexity, where N is the number of sequences and L is their length. There are improvements in selection of heuristics, and close attention is paid to implementation details. This algorithm enables high throughput applications to achieve good accuracy.

ProbCons

ProbCons is another method for approaching multiple alignment. Alignment generation can be directly modeled as a first order Markov process involving state emissions and transitions. ProbCons uses the maximum expected accuracy alignment method, and uses probabilistic consistency as a scoring function. The parameters for the model are obtained using unsupervised maximum likelihood methods. Finally, the algorithm incorporates multiple sequence information for scoring pairwise alignments.

ProbCons uses a basic three state HMM to model alignment (Fig. 7). The HMM has one match state and two gap states, one in each sequence.

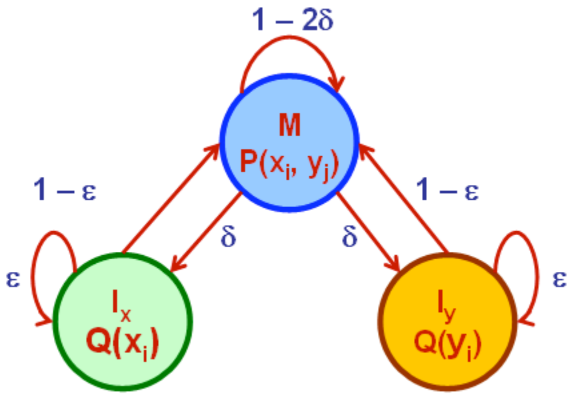


Fig. 7 Simple Three State Pair HMM (Batzoglou, 2005).

ProbCons uses a technique called Maximum Expected Accuracy, instead of the Viterbi which is usually used on HMMs to perform sequence alignment. The Viterbi algorithm picks a single alignment with the highest chance of being completely correct, which is analogous to Needleman-Wunch. It finds the maximum probability alignment, returning a single alignment with the maximum probability of being correct. Maximum expected accuracy picks the alignment with the highest number of correct predictions instead.

ProbCons uses probabilistic consistency, which is similar to the technique we saw earlier used by T-Coffee. This method of using a third sequence for a pairwise alignment helps clear up ambiguities that arise from probabilistic analysis.

Posterior decoding is another technique used in ProbCons. In the two sequence case, posterior decoding amounts to defining a position-dependent substitution matrix and running Needleman Wunsch on it without gap penalties. The unweighted sum-of-pairs technique without gap penalties is used in ProbCons technique is used to compute posterior decoding matrices for multiple sequences.

Here is the summary of the ProbCons algorithm. The first step of the algorithm is computing the posterior-probability matrices. For every pair of sequences x, y, the probability that letters x_i, x_j are paired in an alignment of x, y that is randomly generated by the model, is computed. The second step is the computation of expected accuracies. The expected accuracy of a pairwise alignment a_{xy} is defined to be the expected number of correctly aligned pairs of letters divided by the length of the shorter sequence. In step three, probabilistic consistency transformation is applied and the scores are re-estimated. A guide tree is then constructed by hierarchical clustering. Progressive alignment is computed. The final step, iterative refinement is applied as necessary. Alignments are randomly partitioned into two groups of sequences and re-aligned.

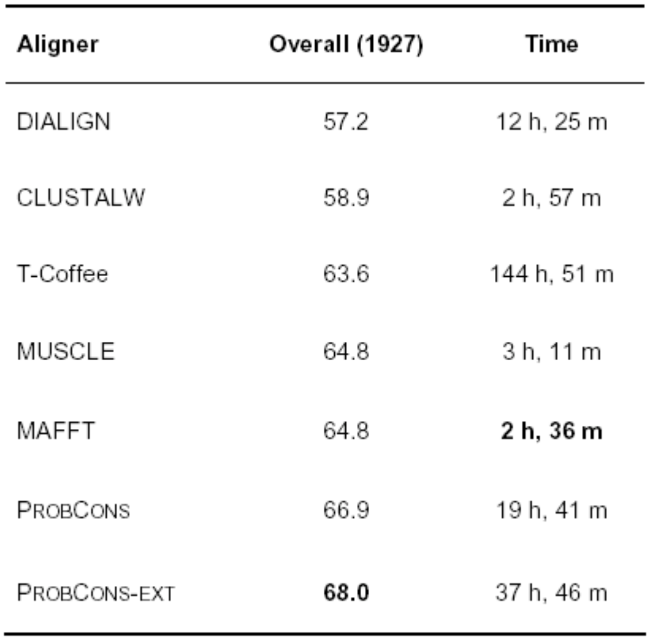
Known alignments are used to train ProbCons by computing the proper HMM parameters. A machine learning technique called expectation maximization is used during the training.

ProbCons is has the best results so far despite the fact that it doesn't incorporate any biological information. It has a slightly longer running time due to comparison of posterior probability matrices, which are done in the first step of the algorithm.

EVALUATION AND DISCUSSION:

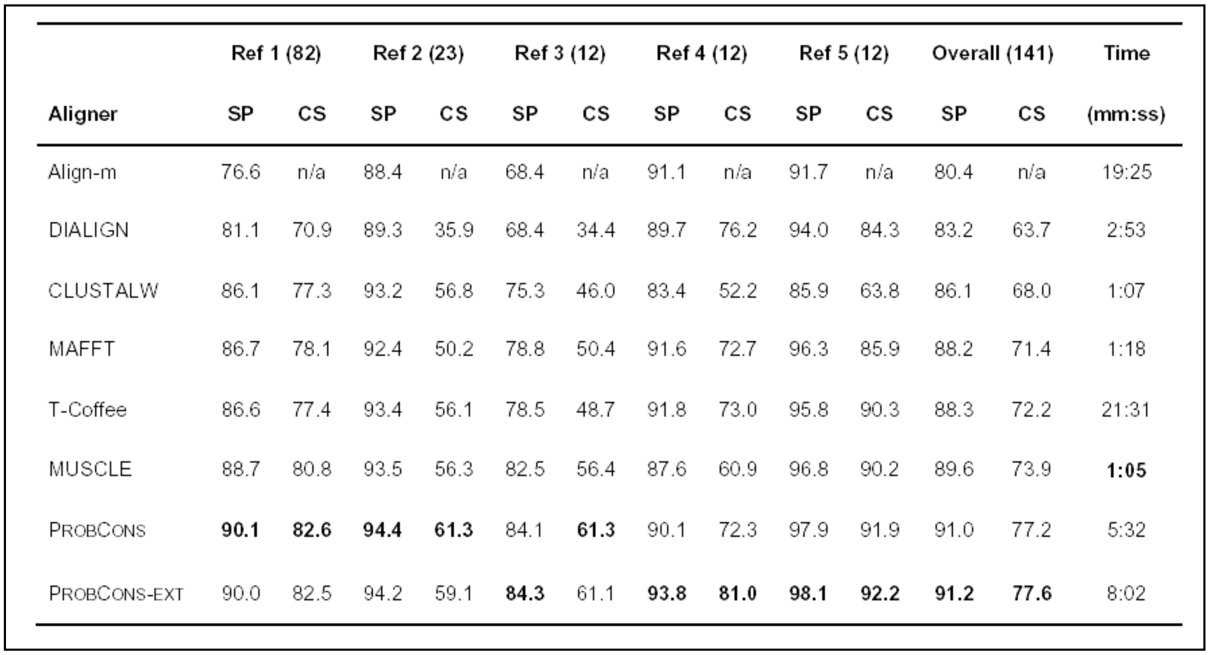
One of the issues that make multiple sequence alignment a difficult problem is that it is very hard to train an aligner. Training requires picking an alignment database and tweaking the parameters of the program until good results are achieved. It is a very time consuming process and overfitting might occur. This is what happens with the ClustalW algorithm. ClustalW was trained on BAliBASE and it performs extremely well on that database as we see from Fig. 9, however it doesn't do as well on other databases (Fig. 8). This problem does not arise in ProbCons, the difference in accuracy between the database on which the aligner was trained and other databases, is not as evident. Looking at Fig. 8

and Fig. 9, which show the comparison of accuracy and running time between quite a few of the methods described in this paper, we see that while ProbCons (especially the extended algorithm which is not discussed in this paper), is the most accurate tool now available, MUSCLE and MAFFT (which is also not discussed in this paper) are the fastest tools. The MUSCLE program has a number of very powerful optimizations which allows it to be as fast as it is. During the development of that system, a lot of attention was paid to implementation details in order to speed up the program and make it as efficient as possible. While T-Coffee is a lot slower than any of the other methods compared, the accuracy of the program itself is quite remarkable. Finally several factors are quite surprising about ClustalW. Although it has been developed over ten years ago, it has decent accuracy and a fairly short running time, both of which are pretty impressive. However, despite the fact that there are much better methods available, it remains to be probably the most widely used protein multiple sequence aligner today.



Aligner	Overall (1927)	Time
DIALIGN	57.2	12 h, 25 m
CLUSTALW	58.9	2 h, 57 m
T-Coffee	63.6	144 h, 51 m
MUSCLE	64.8	3 h, 11 m
MAFFT	64.8	**2 h, 36 m**
PROBCONS	66.9	19 h, 41 m
PROBCONS-EXT	**68.0**	37 h, 46 m

Figure 8. Performance of Aligners on the PREFAB. PREFAB is the Protein Reference Alignment Benchmark. Entries show the average Q, which is equivalent to the Sum-Of-Pairs score achieved by each aligner on all 1927 alignments of the PREFAB database. All scores have been multiplied by a factor of 100. Running times for programs over the entire database are given for each program in hours and minutes. (Do et. al, 2004)



	Ref 1 (82)		Ref 2 (23)		Ref 3 (12)		Ref 4 (12)		Ref 5 (12)		Overall (141)		Time
Aligner	**SP**	**CS**	**SP**	**CS**	**SP**	**CS**	**SP**	**CS**	**SP**	**CS**	**SP**	**CS**	**(mm:ss)**
Align-m	76.6	n/a	88.4	n/a	68.4	n/a	91.1	n/a	91.7	n/a	80.4	n/a	19:25
DIALIGN	81.1	70.9	89.3	35.9	68.4	34.4	89.7	76.2	94.0	84.3	83.2	63.7	2:53
CLUSTALW	86.1	77.3	93.2	56.8	75.3	46.0	83.4	52.2	85.9	63.8	86.1	68.0	1:07
MAFFT	86.7	78.1	92.4	50.2	78.8	50.4	91.6	72.7	96.3	85.9	88.2	71.4	1:18
T-Coffee	86.6	77.4	93.4	56.1	78.5	48.7	91.8	73.0	95.8	90.3	88.3	72.2	21:31
MUSCLE	88.7	80.8	93.5	56.3	82.5	56.4	87.6	60.9	96.8	90.2	89.6	73.9	**1:05**
ProbCons	**90.1**	**82.6**	**94.4**	**61.3**	84.1	**61.3**	90.1	72.3	97.9	91.9	91.0	77.2	5:32
ProbCons-ext	90.0	82.5	94.2	59.1	**84.3**	61.1	**93.8**	**81.0**	**98.1**	**92.2**	**91.2**	**77.6**	8:02

Figure 9. Performance of Aligners on the BAliBASE. BALiBASE is the Benchmark Alignments Database. Columns of this table show the average sum-of-pairs (SP) and column scores (CS) achieved by each aligner for each of the five BAliBASE references. All scores have been multiplied by a factor of 100. The number of sequences in each reference is specified in parentheses. Overall numbers for the entire database are reported in addition to the total running time of each aligner for all 141 alignments. (Do et. al, 2004)

CONCLUSIONS:

Currently, protein multiple sequence alignment remains a very interesting unsolved research problem. Even though ProbCons has the best results so far, the algorithm can also be improved on. A fancier HMM can be used instead of the simple three state model. Methods to extend the model making it more expressive within the context of a probabilities model and attempting to incorporate biological knowledge into the system are all improvements that can be made. Presently there is some collaboration between the authors of ProbCons and MUSCLE to enhance the systems and come up with a better protein multiple sequence aligner, which will combine the speed of MUSCLE-based tree construction and the accuracy that comes from using MEA and probabilistic consistency.

REFERENCES:

Batzoglou, Serafim. CS262 Lecture Notes and Slides. Winter 2005.
<http://cs262.stanford.edu>.

Brutlag, Douglas. BIOCHEM218 Lecture Slides. Winter 2005.
<http://biochem218.stanford.edu>.

Do, C.B., Brudno, M., and Batzoglou, S. PROBCONS: Probabilistic Consistency-based Multiple Alignment of Amino Acid Sequences. *Submitted*.

Edgar, R.C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*. **32**(5): 1792-1797.

Notredame, C., Higgins, D.G., and Heringa, J. 2000. T-Coffee: a novel method for multiple sequence alignments. *J. Mol. Biol.* **302**: 205-217.

Thompson, J.D., Higgins, D.G., and Gibson, T.J. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22**: 4673-4680.