PROBCONS: Probabilistic Consistency-based Multiple Alignment of Amino Acid Sequences

Chuong B. Do, Michael Brudno, Serafim Batzoglou

Department of Computer Science
Stanford University
Stanford, CA 94305
{chuongdo, brudno, serafim}@cs.stanford.edu

Abstract

Obtaining an accurate multiple alignment of protein sequences is a difficult computational problem for which many heuristic techniques sacrifice optimality to achieve reasonable running times. The most commonly used heuristic is progressive alignment, which merges sequences into a multiple alignment by pairwise comparisons along the nodes of a guide tree. To improve accuracy, consistency-based methods take advantage of conservation across many sequences to provide a stronger signal for pairwise comparisons. In this paper, we introduce the concept of *probabilistic consistency* for multiple sequence alignments. We also present PROBCONS, an HMM-based protein multiple sequence aligner, based on an approximation of the probabilistic consistency objective function. On the BAliBASE benchmark alignment database, PROBCONS demonstrates a statistically significant improvement in accuracy compared to several leading alignment programs while maintaining practical running times. Source code and program updates are freely available under the GNU Public License at http://probcons.stanford.edu/.

1 Introduction

Protein multiple alignments display patterns of sequence conservation by organizing homologous amino acids across different protein sequences in columns (see Figure 1). As sequence similarity often implies functional similarity, protein sequence comparisons have been crucial in many bioinformatics applications, including structure prediction (Jones 1999), phylogenetic analysis (Phillips et al. 2000), identification of conserved domains (Attwood 2002), and characterization of protein families (Sonnhammer et al. 1998). However, when the proportion of identities among amino acid matches falls below 30%, called the 'twilight zone' of protein alignments, the accuracies of most automatic sequence alignment methods drop considerably (Rost 1999, Thompson et al. 1999b). As a result, alignment quality is often the limiting factor in comparative modeling studies (Jaroszewski et al. 2002).

Progressive alignment approaches, which assemble complete multiple alignments by successive hierarchical pairwise steps, are commonly used for reasons of algorithmic efficiency (Feng and Doolittle 1987). Unfortunately, such strategies are highly prone to errors at early stages of the alignment. To combat this, consistency-based alignment techniques use shared homology with outgroup sequences to distinguish between evolutionary and coincidental sequence similarity, thus improving the quality of pairwise comparisons (Gotoh 1990, Morgenstern et al. 1998, Notredame et al. 2000).

```
---NAYCDEECKKG---AESGKCWY
-YDNAYCDKLCKDK--KADSGYCYW
-TAAGYCNTECTLK--KGSSGYCAW
LGKNDYCNRECRMKHRGGSYGYCYG
--GNEGCNKECKSY--GGSYGYCWT
```

Figure 1: A five-sequence multiple alignment example.

In this paper, we present PROBCONS, a protein multiple alignment tool that performs consistency-based progressive alignment while accounting for all suboptimal alignments with posterior-probability–based scoring. Other features of the program include the use of double affine insertion penalties, guide tree calculation via semi-probabilistic hierarchical clustering, optional iterative refinement, and unsupervised Expectation-Maximization (EM) training of gap parameters. On the BAliBASE (Thompson et al. 1999a) reference dataset, PROBCONS gives statistically significant improvements in alignment quality when compared to several leading alignment tools, including CLUSTALW (Thompson et al. 1994), DIALIGN (Morgenstern et al. 1998), and T-Coffee (Notredame et al. 2000), while maintaining comparable running times. Source code and updates for our system are publicly available under the GNU Public License at http://probcons.stanford.edu/.

1.1 Pair-HMMs and alignment

Given two sequences $x = x_1 \ldots x_{|x|}$ and $y = y_1 \ldots y_{|y|}$, a pairwise alignment indicates positions of each sequence that are considered to be functionally or evolutionarily related. Traditional edit-distance–based methods score an alignment as the sum of similarity values for aligned residues and length-dependent gap penalties for unaligned positions (Needleman and Wunsch 1970, Smith and

Waterman 1981). The similarity of specific residue pairs may be quantified with log-odds scores (Henikoff and Henikoff 1992), and affine gap parameters are generally estimated empirically (Vingron and Waterman 1994). For two sequences of length L, an optimal alignment may be computed in $O(L^2)$ time (Gotoh 1982) and $O(L)$ space (Myers and Miller 1988) via dynamic programming.

Pair hidden Markov models (HMMs) provide a natural probabilistic interpretation of alignment as a stochastic process in which two sequences are jointly generated according to a first-order Markov process. Typical pair-HMMs for alignment contain three states, including one match state that emits letters from x and y simultaneously according to a joint distribution for residue pairs, and two insert states that emit letters from one sequence at a time (Allison et al. 1992). A sequence of states that generates x and y corresponds uniquely to an alignment. Finally, the highest probability alignment can be computed with the Viterbi algorithm (Durbin et al. 1998).

Modeling an alignment as a pair-HMM confers several advantages. The alignment parameters obtain an intuitive probabilistic interpretation and can be trained using standard supervised or unsupervised maximum likelihood methods. Pair-HMMs, moreover, allow one to compute the *posterior probability*, $P(x_i \sim y_j \mid x, y)$, that particular positions x_i and y_j of the two sequences x and y, respectively, will be matched in the final alignment.

1.2 Consistency-based alignment

Scoring a multiple alignment in a probabilistically rigorous and biologically motivated manner, and finding the optimal alignment once a scoring scheme has been specified, are not straightforward tasks. In practice, the ad hoc sum-of-pairs measure, which combines the projected pairwise scores for aligning all pairs of sequences (Carrillo and Lipman 1988), and its weighted variants (Altschul et al. 1989) are commonly used for scoring. Direct application of dynamic programming is too inefficient in the multiple sequence case, so many heuristic strategies use progressive alignment based on an evolutionary tree (Feng and Doolittle 1987), or iterative approaches including genetic algorithms (Notredame and Higgins 1996), simulated annealing (Kim et al. 1994), alignment to a profile HMM (Eddy 1995), or greedy assemblage of multiple segment-to-segment comparisons (Morgenstern et al. 1996). Progressive methods such as CLUSTALW are the most popular but are prone to errors in early stages of alignment. To combat this, post-processing steps such as iterative refinement (Gotoh 1996) may be applied.

Consistency-based schemes take the alternative view that "prevention is the best medicine." Note that for any multiple alignment, the induced pairwise alignments are necessarily consistent—i.e., given a multiple alignment of x, y, and z, if position x_i aligns with position z_k and position z_k aligns with y_j in the projected x-z and z-y alignments, then x_i must align with y_j in the projected x-y alignment. Consistency-based techniques apply this principle in reverse, using alignments to intermediate sequences as evidence to guide the pairwise alignment of x and y.

Gotoh (1990) first introduced consistency to identify anchor points for reducing the search space of a multiple alignment. A mathematically elegant reformulation of consistency in terms of boolean matrix multiplication was later given by Vingron and Argos (1991) and implemented in the program MALI, which builds multiple alignments from dot matrices (Vingron and Argos 1989). An alternative formulation of consistency was employed in the DIALIGN tool, which finds ungapped local alignments via segment-to-segment comparisons, determines new weights for these alignments using consistency, and assembles them into a multiple alignment by a greedy selection procedure (Morgenstern et al. 1996).

More recently, Notredame et al. (1998), introduced COFFEE, a new consistency-based objective function for scoring residue pairs in a pairwise alignment. In this approach, an alignment library is computed by merging consistent CLUSTALW global and LALIGN (Huang and Miller 1991) local pairwise alignments to form three-way alignments, which are assigned weights by percent identity. The score for aligning x_i to y_j is calculated by summing the weights of all alignments in the library containing that aligned residue pair. The program T-Coffee, which implements this objective function using progressive alignment based on pairwise maximum weight trace computations (Kececioglu 1993), has demonstrated superior accuracy on the BAliBASE test suite over competing methods, including CLUSTALW, DIALIGN, and PRRP (Gotoh 1996).

2 Computing pairwise alignments

To score an alignment between two sequences by using evidence from a third sequence, existing consistency-based methods rely on various heuristics for weighting consistency-derived information. DIALIGN uses the weight of the overlap shared between two consistent diagonals as a measure of consistency strength; T-Coffee uses alignment percent identity to determine weights for each residue pairing from two consistent alignments. No existing consistency-based alignment methods, however, use pair-HMM posterior probabilities, which give a per-position measure of alignment reliability that takes into account the effect of *all possible suboptimal alignments*. The combination of consistency and posterior-derived weights is the basis for probabilistic consistency.

In this section, we introduce the concept of probabilistic consistency with respect to calculation of two sequence alignments and describe an exact $O(L^3)$ method for computing an objective function based on this concept using triple-HMMs, where L is the total length of the input sequences. As this method is computationally expensive in practice, we present an efficient $O(L^2)$ approximation of our objective function. In Section 3, we extend this approximation to the calculation of multiple alignments.

2.1 Definitions

As before, let x and y be two proteins represented as character strings in which x_i is the ith amino acid of x. Given a pair-HMM, M, such as the one described in Section 1.1, each possible sequence of states for generating the sequences x and y corresponds to a *pairwise global alignment*. For an alignment a of x and y, we say $(i, j) \in a$ if positions x_i and y_j are aligned to each other in a.

Let $A_{x,y}$ denote the set of all possible pairwise global alignments of two sequences x and y, and let $P(A_{x,y})$ be a probability distribution over all possible alignments, as specified by M. We denote the correct or true alignment of x and y (in an evolutionary or functional sense) as a^*. In the algorithms which follow, we assume that such an alignment a^* exists and that a^* was drawn from the distribution $P(A_{x,y})$.

Most alignment schemes build an "optimal" pairwise alignment by finding the highest probability alignment using the Viterbi algorithm. The resulting alignment reflects a global maximum of the probability distribution $P(A_{x,y})$ and may be highly sensitive to the exact parameters used in computing the alignment. In this work, we explore an alternative strategy that maximizes not the probability of an alignment but rather its *accuracy*, defined as

$$accuracy(a) = \frac{\sum 1\{(i, j) \in a^*\}}{\min\{|x|, |y|\}},$$

where the indicator notation $1\{condition\}$ is equal to 1 if *condition* is true and 0 otherwise.

In general, however, a^* is not known, so we instead maximize the expected accuracy of the reported alignment. To do this, we define the *posterior probability* of a match between x_i and y_j occurring in the true alignment as

$$P(x_i \sim y_j \mid x, y) = \sum_{a \in A_{x,y}} P(a) 1\{(i, j) \in a\}.$$

Note that under the assumption that the true alignment a^* is drawn from the distribution $P(A_{x,y})$, the posterior probability may be considered the expectation value that a particular residue pairing is correct.

Theorem 2.1: *The expected accuracy of an alignment may be computed as*

$$E\{accuracy(a)\} = \frac{\sum P(x_i \sim y_j \mid x, y)}{\min\{|x|, |y|\}}.$$

The proof of the theorem follows easily by applying linearity of expectations to the definition of accuracy.

We note that in many applications of HMMs, Viterbi is a superior choice to maximizing expected accuracy, because the resulting parse from the latter approach may fail to respect the relationships between highly correlated states in the HMM topology. In such a case, the highest expected accuracy parse will optimize local accuracy while creating a globally inconsistent parse. In the case of alignment, however, this is not the case as dependencies between states are not as prevalent; thus, per-position accuracy is a reasonable quantity to maximize.

2.2 Probabilistic consistency

In the process described above, computing the optimal expected accuracy involves finding the alignment for which the sum of the $P(x_i \sim y_j \mid x, y)$'s is maximal. Intuitively, the $P(x_i \sim y_j \mid x, y)$'s provide a quality score for determining the desirability of various position matches.

When a third homologous sequence z is available, however, consistency provides a means for obtaining a better estimate for the quality of aligning x_i and y_j. Given $P(A_{x,y,z})$, a joint distribution over all three sequence alignments of sequences x, y, and z, rather than using $P(x_i \sim y_j \mid x, y)$ values as quality scores, one may use the marginalized probabilities $P(x_i \sim y_j \mid x, y, z)$. We refer to the re-estimation of pairwise alignment probabilities based on three-sequence information as *probabilistic consistency*.

Note that the above computation may performed using a three-sequence HMM, for which evaluating posterior probabilities involves an $O(L^3)$ calculation using the forward and backward algorithms. For most sequences, however, a cubic running time is unacceptable. Thus, we employ the following heuristic "factorization":

$$P(x_i \sim y_j \mid x, y, z) \approx \sum_k P(x_i \sim z_k \mid x, z) P(z_k \sim y_j \mid z, y).$$

Admittedly, the independence assumptions required for such a transformation are in principle unjustified and do not account for alignments of x_i and y_j to gaps in z. Nevertheless, the transformation works well in practice, as seen in Section 4. From a computational perspective, the running time is reduced to approximately $O(L^2)$—normally, a position x_i will align to only a few positions z_k, so the $P(x_i \sim z_k \mid x, z)$ values may be represented as a sparse matrix (and similarly for $P(z_k \sim y_j \mid z, y)$); thus, computing each $P(x_i \sim y_j \mid x, y, z)$ takes approximately constant time.

3 Computing multiple alignments

Thus far, we have described how to use pair-HMM–derived posterior probabilities to compute pairwise alignments of two sequences that maximize expected accuracy. Furthermore, we have introduced a heuristic approximation for obtaining consistency-based quality scores for aligning particular residue pairs when a third homologous sequence is present. In this section, we extend the pairwise alignment model to multiple alignment under a progressive scheme.

3.1 Progressive alignment

In the pairwise alignment case, an optimal expected accuracy alignment is calculated by applying the maximum weight trace algorithm (Kececioglu 1993) to a matrix consisting of the $P(x_i \sim y_j \mid x, y)$ values; a related scheme is used in the T-Coffee program.

For the multiple sequence case, we rely on a progressive alignment scheme using the guide tree calculated by hierarchical clustering of sequences by the method given in Section 3.2. Initially, progressive alignment proceeds by assigning each sequence to its corresponding leaf in the tree. If a node has exactly two leaves as children, then the sequences assigned to the children are aligned, the alignment is assigned to the node, the leaves are removed, and the process repeats until only a single node remains and all sequences have been aligned.

When aligning groups of sequences, we use a *sum-of-pairs* scheme in which the score of a multiple alignment is given by summing the expected accuracy for each possible projection of the multiple alignment to two sequences. Thus, we can find the optimal expected accuracy alignment of a group of sequences by a straightforward extension of the pairwise dynamic programming computation.

3.2 Guide tree calculation

Guide trees are computed in a greedy hierarchical manner. Given a set S of sequences to be aligned, we denote the expected accuracy for aligning two sequences x and y as $E(x, y)$. Initially, each sequence is placed in its own cluster. Then, the two clusters x and y with the highest expected accuracy are merged to form a new cluster xy; we then define the expected accuracy of aligning xy with any other cluster z as $E(x, y)(E(x, z) + E(y, z)) / 2$. This process is repeated until only a single cluster remains.

3.3 The probabilistic consistency transformation

To align two sequences x and y given a set of multiple sequences, S, we would ideally estimate $P(x_i \sim y_j \mid S)$. In practice, we use the following heuristic decomposition:

$$P(x_i \sim y_j \mid S) \leftarrow \frac{1}{|S|}\sum_{z \in S}\sum_{k=1}^{|z|} P(x_i \sim z_k \mid x, z) P(z_k \sim y_j \mid z, y)$$

where we set $P(x_i \sim x_j \mid x)$ to 1 if $i = j$ and 0 otherwise.

In this sense, the approximate probabilistic consistency calculation may be viewed as a transformation that, given a set of all-pairs pairwise posterior probabilities, produces a new set of all-pairs pairwise posterior probabilities that have been adjusted to account for a single intermediate sequence. By *iterated applications* of the transformation, then, we can approximate the effect of accounting for more than one intermediate sequence at a time.

3.4 PROBCONS aligner

We implemented PROBCONS, an alignment tool based on the $O(L^2)$ approximate probabilistic consistency technique detailed above. Given sequences x and y, the probability distribution over all possible global alignments is specified by a pair-HMM with states for emitting matched pairs of residues (M), short insertions in one sequence (I_1^x and I_1^y), and long insertions in one sequence (I_2^x and I_2^y). The allowed transitions and their associated probabilities are given in Figure 2.

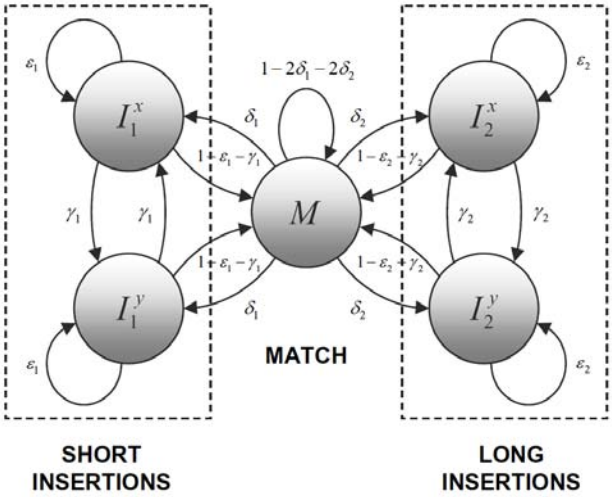

Figure 2: PROBCONS HMM topology. The transition parameters correspond to the probabilities of starting a new insertion (δ_1 and δ_2), continuing an insertion (ε_1 and ε_2), and switching between inserted sequences (γ_1 and γ_2). The initial probabilities for starting in a short or long insertion are given by π_1 and π_2 (not shown).

PROBCONS begins by computing all-pairs pairwise posterior probabilities, followed by two iterations of the probabilistic consistency transformation. One optional iteration of EM training over the sequences to be aligned is used to generate sequence-specific training parameters before the posteriors are calculated. After computing a guide tree as described above, a multiple alignment is generated using the progressive alignment procedure.

To further improve the quality of this alignment, 100 rounds of *iterative refinement* are applied; in this procedure, sequences of the multiple alignment are randomly partitioned into two groups (where each sequence is placed in the first group with probability 0.5), and the groups are realigned using the probabilistic consistency sum-of-pairs objective function. As the original multiple alignment can always be regenerated, the alignment score is guaranteed not to decrease.

4 Evaluation

To test the empirical performance of the PROBCONS aligner, we used the online BAliBASE 2.0 benchmark alignment database, a collection of 141 reference protein alignments consisting of structural alignments from the FSSP (Holm and Sander 1996) and HOMSTRAD (Mizuguchi et al. 1998) databases and hand-constructed alignments from the literature. The database is organized into five reference sets: Reference 1 consists of few equidistant sequences of similar length; Reference 2, families of closely-related sequences with up to three distant "orphan" sequences; Reference 3, equidistant divergent families; Reference 4, sequences with large N/C-terminal extensions; and Reference 5, sequences with large internal insertions.

Algorithm	*Ref* 1 (82)		*Ref* 2 (23)		*Ref* 3 (12)		*Ref* 4 (12)		*Ref* 5 (12)		*Overall* (141)		Time
	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	(mm:ss)
CLUSTALW	86.4	78.3	88.9	40.6	75.5	46.8	81.1	50.4	86.1	63.9	85.4	65.9	**1:05**
DIALIGN	81.3	71.4	85.0	27.9	68.6	34.8	91.2	81.9	94.1	84.5	82.8	63.2	3:04
T-Coffee	86.8	77.9	88.6	38.9	78.8	49.5	91.9	74.9	96.0	90.5	87.6	69.9	24:02
PROBCONS	**90.3**	**83.2**	**91.5**	**48.9**	**85.1**	**63.1**	**95.2**	**85.7**	**98.2**	**92.4**	**91.1**	**76.9**	8:26

Table 1: Comparison of BALIBASE performance for DIALIGN, CLUSTALW, T-Coffee, and PROBCONS. The time required to run on the entire BAliBASE data is reported. The best result in each column is shown in bold.

Algorithm	CLUSTALW		DIALIGN		T-Coffee		PROBCONS	
	SP	CS	SP	CS	SP	CS	SP	CS
% unique best alignment	12.1%	9.9%	3.5%	5.7%	14.2%	14.2%	**51.8%**	**46.1%**
% best alignment	19.9%	21.3%	13.5%	19.1%	30.5%	34.0%	**68.1%**	**66.7%**

Table 2: Percentage of alignments in which each method produced the (1) unique best alignment or (2) the best alignment (two or more methods achieved the same highest accuracy). The best results in each row are shown in bold.

s	*c*	*ir*	*em*	*Ref* 1 (82)		*Ref* 2 (23)		*Ref* 3 (12)		*Ref* 4 (12)		*Ref* 5 (12)		*Overall* (141)		Time
				SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	(mm:ss)
1	0	0	0	87.7	79.3	89.1	36.2	83.4	52.3	86.5	63.0	96.2	85.9	88.2	69.1	2:45
2	0	0	0	87.8	79.3	89.8	41.5	83.3	52.6	86.6	63.9	95.2	83.4	88.2	69.9	5:08
2	1	0	0	89.1	81.4	91.2	47.3	85.5	62.3	90.5	73.2	97.5	90.5	90.0	74.3	5:29
2	2	0	0	89.4	81.8	91.5	48.9	85.1	63.1	90.5	73.2	98.2	92.4	90.2	75.0	5:54
2	2	100	0	90.3	83.2	91.5	48.9	85.1	63.1	95.2	85.7	98.2	92.4	91.1	76.9	8:26
2	2	100	1	90.6	83.5	91.7	49.5	85.3	63.8	95.2	85.7	98.2	92.4	91.4	77.2	14:14

Table 3: Comparison of BALIBASE performance for PROBCONS variants. The four parameters varied over these runs include: *s*, the number of insertion state pairs in the HMM topology; *c*, the number of consistency transformation applied; *ir*, the number of rounds of iterative refinement via randomized partitioning; and *em*, the number of unsupervised EM iterations used to train sequence-specific parameters for each set before aligning.

BAliBASE *core blocks*, regions for which reliable alignments are known to exist, comprise 58% of all residues in the database. To assess alignment quality, we used the ALN_COMPARE program (Notredame et al. 2000), which scores alignments against a reference according to both the *sum-of-pairs score* (SP), the percentage of aligned core block residue pairs which are also aligned in the reference, and the *column score* (CS), the percentage of aligned columns that are also aligned in the core blocks of the reference.

Emission probabilities for the PROBCONS HMM were estimated from the statistics used in generating the BLOSUM62 scoring matrix (Henikoff and Henikoff 1992). The remaining parameters, consisting of six transition (δ_1, ε_1, γ_1, δ_2, ε_2, γ_2) and two initial distribution probabilities (π_1, π_2) were obtained via unsupervised EM through two-fold cross validation.

4.1 Comparison against existing tools

We compared the results of the PROBCONS aligner to that of CLUSTALW 1.83, DIALIGN 2.2.1, and T-Coffee 1.37; default settings were used for all programs. From the results above (see Tables 1 and 2), PROBCONS shows a clear advantage over the other methods. Applying the Wilcoxon matched-pairs signed-ranks test over all 141 alignments indicated significant improvement over the other three methods (with $p < 10^{-6}$ for both SP and CS measures). In other experiments (results omitted for space reasons), we found no significant difference between the full $O(L^3)$ and the approximate $O(L^2)$ probabilistic consistency alignment procedures.

4.2 Comparison of PROBCONS variants

We also ran tests which showed that (1) using two insertion state pairs instead of one, (2) using iterative refinement, and (3) performing EM on each set before aligning to estimate better sequence-specific parameters all gave improvements in alignment quality (see Table 3).

5 Discussion

In this paper, we developed a framework for probabilistic consistency of alignments and implemented these ideas in the PROBCONS protein multiple alignment tool. Features of the program include (1) posterior probability scoring to account for suboptimal alignments, (2) the use of two insertion state pairs to model short and long insertions separately, (3) guide tree construction via expected accuracies rather than phylogenetic distance, (4) optimization of alignment expected accuracy rather than probability, (4) iterative refinement via a randomized

partitioning strategy, and (5) unsupervised EM training for obtaining parameters. As demonstrated, PROBCONS provides a dramatic improvement in alignment quality over existing methods—achieving the highest scores on the BAliBASE alignment benchmark of any currently known alignment program—while maintaining practical running times. Source code and updates to the program are freely downloadable from http://probcons.stanford.edu/.

The potential applications of probabilistic consistency are not limited to protein alignments. Similar techniques may be used for DNA alignment, motif finding, and RNA structural prediction. In general, methods in which multiple sources of information are decomposed into pairwise relations are potential targets for this technique. We will continue developing these ideas, to unlock the full potential of the probabilistic consistency methodology.

Acknowledgments

We thank Mahathi Mahabhashyam and Sandhya Kunnatur for help in program development. CBD was partly supported by a Siebel Fellowship. MB was partly supported by an NSF Graduate Fellowship. SB and CBD were supported in part by NSF grant EF-0312459.

References

Allison, L., Wallace, C.S., Yee, C.N. 1992. Finite-state models in the alignment of macromolecules. *J Mol Evol* 35(1):77-89.

Altschul, S.F., Carroll, R.J., and Lipman, D.J. 1989. Weights for data related by a tree. *J Mol Biol* 207:647-653.

Attwood, T.K. 2002. The PRINTS database: a resource for identification of protein families. *Brief Bioinform* 3(3):252-263.

Carrillo, H. and Lipman, D. 1988. The multiple sequence alignment problem in biology. *SIAM J Appl Math* 48:1073-1082.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. 1998. *Biological Sequence Analysis*. Cambridge UP, Cambridge.

Eddy, S.R. 1995. Multiple alignment using hidden Markov models. In *Proc 3rd International Conference on Intelligent Systems in Molecular Biology*, 114-120. Menlo Park, Calif.: AAAI Press.

Feng, D.F., and Doolittle, R.F. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol* 25:351-360.

Gotoh, O. 1982. An improved algorithm for matching biological sequences. *J Mol Biol* 162:705-708.

Gotoh, O. 1990. Consistency of optimal sequence alignments. *Bull Math Biol* 52:509-525.

Gotoh, O. 1996. Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J Mol Biol* 264:823-838.

Henikoff, S., and Henikoff, J.G. 1992. Amino acid substitution matrices from protein blocks. *Proc Nat Acad Sci USA* 89:10915-10919.

Holm, L., and Sander, C. 1996. Mapping the protein universe. *Science* 273:595-602.

Huang, X., and Miller, W. 1991. A time-efficient, linear space local similarity algorithm. *Adv Appl Math* 12:337-357.

Jaroszewski, L., Li, W., Godzik, A. 2002. In search for more accurate alignments in the twilight zone. *Prot Sci* 11(7):1702-1713.

Jones, D.T. 1999. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol* 292(2):195-202.

Kececioglu, J. 1993. The maximum weight trace problem in multiple sequence alignment. In *Proc 4th Symposium on Combinatorial Pattern Matching*, Springer-Verlag Lecture Notes in Computer Science 684:106-119.

Kim, J., Pramanik, S., and Chung, M.J. 1994. Multiple sequence alignment using simulated annealing. *Comput Appl Biosci* 10(4):419-426.

Mizuguchi, K., Deane, C.M., Blundell, T.L., and Overington, J.P. 1998. HOMSTRAD: a database of protein structure alignments for homologous families. *Prot Sci* 7:2469-2471.

Morgenstern, B., Dress, A., and Werner, T. 1996. Multiple DNA and protein sequence alignment based on segment-to-segment comparison. *Proc Nat Acad Sci USA* 93:12098-12103.

Morgenstern, B., Frech, K., Dress, A., and Werner, T. 1998. DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics* 14:290-294.

Myers, E.W. and Miller, W. 1988. Optimal alignments in linear space. *Comput Appl Biosci* 4:11-17.

Needleman, S.B., and Wunsch, C.D. 1970. A general method applicable to the search for similarities in amino acid sequence of two proteins. *J Mol Biol* 48:443-453.

Notredame, C., and Higgins, D.G. 1996. SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res* 24:1515-1524.

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

Notredame, C., Holm, L., and Higgins, D.G. 1998. COFFEE: An objective function for multiple sequence alignments. *Bioinformatics* 14(5):407-422.

Phillips, A., Janies, D., and Wheeler, W. 2000. Multiple sequence alignment in phylogenetic analysis. *Mol Phylogenet Evol* 16(3):317-330.

Rost, B. 1999. Twilight zone of protein sequence alignments. *Protein Eng* 12(2):85-94.

Smith, T.F., and Waterman, M.S. 1981. Identification of common molecular subsequences. *J Mol Biol* 147:195-197.

Sonnhammer, E.L.L., Eddy, S.R., Birney, E., Bateman, A., and Durbin, R. 1998. Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res* 26(1):320-322.

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.

Thompson, J.D., Plewniak, F., and Poch, O. 1999a. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics* 15(1):87-88.

Thompson, J.D., Plewniak, F., and Poch, O. 1999b. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res* 27(13):2682-2690.

Vingron, M., and Argos, P. 1991. Motif recognition and alignment for many sequences by comparison of dot matrices. *J Math Biol* 218:34-43.

Vingron, M., and Argos, P. 1989. A fast and sensitive multiple sequence alignment algorithm. *Comput Appl Biosci* 5(2):115-121.

Vingron, M, and Waterman, M.S. 1994. Sequence alignment and penalty choice: review of concepts, case studies and implications. *J Mol Biol* 235:1-12.