Sequence Analysis

Probalign: Multiple sequence alignment using partition function posterior probabilities

Usman Roshan[1*] and Dennis R. Livesay[2]

[1] Department of Computer Science, New Jersey Institute of Technology

[2] Department of Computer Science and Bioinformatics Research Center, University of North Carolina at Charlotte

ABSTRACT

Motivation: The maximum expected accuracy optimization criterion for multiple sequence alignment uses pairwise posterior probabilities of residues to align sequences. The partition function methodology is one way of estimating these probabilities. Here, we combine these two ideas for the first time to construct maximal expected accuracy sequence alignments.

Results: We bridge the two techniques within the program Probalign. Our results indicate that Probalign alignments are generally more accurate than other leading multiple sequence alignment methods (i.e., Probcons, MAFFT, and MUSCLE) on the BAliBASE 3.0 protein alignment benchmark. Similarly, Probalign also outperforms these methods on the HOMSTRAD and OXBENCH benchmarks. Probalign ranks statistically significantly highest (P-value < 0.005) on all three benchmarks. Deeper scrutiny of the technique indicates that the improvements are largest on datasets containing N/C terminal extensions and on datasets containing long and heterogeneous length proteins. These points are demonstrated on both real and simulated data. Finally, our method also produces accurate alignments on long and heterogeneous length datasets containing protein repeats. There, alignment accuracy scores are at least 10% and 15% higher than the other three methods when standard deviation of length is at least 300 and 400 respectively.

Availability: Open source code implementing Probalign as well as for producing the simulated data, and all real and simulated data are freely available from http://www.cs.njit.edu/usman/probalign

Contact: usman@cs.njit.edu

1 INTRODUCTION

Protein sequence alignment is likely the most commonly used task in bioinformatics (Notredame *et al.*, 2002). Applications include detecting functional regions in proteins (La *et al*., 2005) and reconstructing complex evolutionary histories (Notredame *et al.*, 2002; Durbin *et al.*, 1998). Techniques for constructing accurate alignments are therefore of great interest to the bioinformatics community. The bioinformatic literature is filled with many alignment tools, e.g., ClustalW (Thompson *et al.*, 1994), Dialign (Subramanian *et al.,* 2005), T-Coffee (Notredame *et al.,* 2000), Probcons (Do *et al.*, 2005), MUSCLE (Edgar, 2004), and MAFFT (Katoh *et al.*, 2005). In terms of accuracy, recent comparative studies (Do *et al.,* 2005; Katoh *et al.*, 2005; Edgar 2004) place MAFFT and Probcons among the very top performing sequence alignment methods.

Given the importance of multiple sequence alignment, several protein alignment benchmarks have been created for unbiased accuracy assessment of alignment quality. Of these, BAliBASE (Thompson et al., 1999; Bahr *et al.*, 2001; Thompson *et al.*, 2005) is by far the most commonly used. The BAliBASE benchmark alignments are computed using superimposition of protein structures. To date Probcons v1.1 and MAFFT v5.851 are the most accurate on BAliBASE, whereas MUSCLE is among the fastest on these benchmarks (see Do *et al.,* 2005; Edgar 2004; and Katoh *et al.,* 2005 for recent studies).

MUSCLE is a sum-of-pairs optimizer, which uses the log expectation score for aligning profiles of sequences. It is among the fastest alignment programs in the literature. Additionally, the accuracy of the MUSCLE alignments is generally quite good. MAFFT is based upon Fast Fourier Transforms; though, the latest version, combines different optimization criteria that evaluate consistency between multiple and pairwise alignments. Probcons computes the maximal expected accuracy alignment instead of the usual maximum sum-of-pairs or the Viterbi alignment (Durbin *et al.,* 1998). The expected accuracy of an alignment is based upon posterior probabilities of residues (Durbin *et al.,* 1998; Miyazawa 1995). Probcons computes these probabilities using a Hidden Markov Model (HMM) for pairwise sequence alignment. The HMM parameters are learned using unsupervised learning on the BAliBASE 2.0 benchmark.

In this investigation, we bridge two important bioinformatic techniques (for the first time) in an effort to produce more accurate multiple sequence alignments. The first approach estimates amino acid posterior probabilities from the partition function of alignments (as described by Miyazawa 1995). The second computes the maximal expected accuracy alignment (as described originally by Durbin *et al.,* 1998) after applying the probability consistency transformation of Probcons (Do *et al.,* 2005). The new method, which we call Probalign, generally produces statistically significantly better alignments than the state-of-the-art on the BAliBASE 3.0, HOMSTRAD, and OXBENCH benchmarks. The improvements are largest when datasets of variable and long length sequences are considered.

2 METHODS

Posterior probabilities and maximal expected accuracy alignment

[*] To whom correspondence should be addressed.

Most alignment programs compute an optimal sum-of-pairs alignment or a maximum probability alignment using the Viterbi algorithm (Durbin *et al.*, 1998). An alternative approach is to search for the maximum expected accuracy alignment (Durbin *et al.,* 1998; Do *et al.*, 2005). The expected accuracy of an alignment is based upon the posterior probabilities of aligning residues in two sequences.

Consider sequences x and y and let a^* be their true alignment. Following the description in (Do *et al.*, 2005) the posterior probability of residue x_i aligned to y_j in a^* is defined as

$$P(x_i \sim y_j \in a^* | x, y) = \sum_{a \in A} P(a|x,y) \mathbf{1}\{x_i \sim y_j \in a\} \qquad (1)$$

where A is the set of all alignments of x and y and ***1****(expr)* is the indicator function which returns 1 if the expression *expr* evaluates to true and 0 otherwise. $P(a|x,y)$ represents the probability (our belief) that alignment a is the true alignment a^*. This can easily be calculated using a pairwise HMM if all the parameters are known (see Do *et. al.*, 2005). From hereon we represent the posterior probability as $P(x_i \sim y_j)$ with the understanding that it represents the probability of x_i aligned to y_j in the true alignment a^*. Given the posterior probability matrix $P(x_i \sim y_j)$, we can compute the maximal expected accuracy alignment using the following recursion described in Durbin *et al.*, 1998.

$$A(i,j) = \max \begin{Bmatrix} A(i-1, j-1) + P(x_i \sim y_j) \\ A(i-1, j) \\ A(i, j-1) \end{Bmatrix} \qquad (2)$$

Probcons estimates posterior probabilities for amino acid residues using pair HMMs and unsupervised learning of model parameters. It then proceeds to construct a maximal expected accuracy alignment by aligning pairs of sequence profiles along a guide-tree followed by iterative refinement. In this investigation, we examine a different technique of estimating posterior probabilities; we use suboptimal alignments generated using the partition function of alignments.

According to equation (1) as long as we have an ensemble of alignments A with their probabilities $P(a|x,y)$ we can compute the posterior probability $P(x_i \sim y_j)$ by summing up the probabilities of alignments where x_i is paired with y_j . One way to generate an ensemble of such alignments is to use the partition function methodology, which we now describe.

Posterior probabilities by partition function

Amino acid scoring matrices, normally used for sequence alignment, are represented as log-odds scoring matrices (as defined by Dayhoff *et al.*, 1978). The commonly used sum-of-pairs score of an alignment a (Durbin et. al., 1998) is defined as the sum of residue-residue pairs and residue-gap pairs under an affine penalty scheme.

$$S(a) = T \sum_{(i,j) \in a} \ln(M_{ij} \; / \; f_i f_j) + (gap\_penalties) \qquad (3)$$

Here T is a constant (depending upon the scoring matrix), M_{ij} is the mutation probability of residue i changing to j and f_i and f_j are background frequencies of residues i and j. In fact, it can be shown that any scoring matrix corresponds to a log odds matrix (Karlin and Alstchul 1990; Altschul 1993).

Miyazawa 1995 proposed that the probability of alignment a, $P(a)$, of sequences x and y can be defined as

$$P(a) \propto e^{S(a)/T} \qquad (4)$$

where $S(a)$ is the score of the alignment under the given scoring matrix. In this setting one can then treat the alignment score as negative energy and T as the thermodynamic temperature, similar to what is done in statistical mechanics. Analogous to the statistical mechanical framework, Miyazawa 1995 defined the partition function of alignments as

$$Z(T) = \sum_{a \in A} e^{S(a)/T} \qquad (5)$$

where A is the set of all alignments of x and y. With the partition function in hand, the probability of an alignment a can now be defined as

$$P(a,T) = e^{S(a)/T} \; / \; Z(T) \qquad (6)$$

As T approaches infinity all alignments are equally probable, whereas at small values of T, only the nearly optimal alignments have the highest probabilities. Thus, the temperature parameter T can be interpreted as a measure of deviation from the optimal alignment.

The alignment partition function can be computed using recursions similar to the Needleman-Wunsch dynamic algorithm. Let Z^M_{ij} represent the partition function of all alignments of $x_{1..i}$ and $y_{1..j}$ ending in x_i paired with y_j, and $S_{ij}(a)$ represent the score of alignment a of $x_{1..i}$ and $y_{1..j}$. According to equation (5)

$$Z^M_{i,j} = \sum_{a \in A_{ij}} e^{S_{ij}(a)/T} = \left(\sum_{a \in A_{i-1j-1}} e^{S_{i-1,i-1}(a)/T} \right) e^{s(x_i, y_j)/T} \qquad (7)$$

where A_{ij} is the set of all alignments of $x_{1..i}$ and $y_{1..j}$, and $s(x_i,y_j)$ is the score of aligning residue x_i with y_j. The summation in the bracket on the right hand side of equation (7) is precisely the partition function of all alignments of $x_{1..i-1}$ and $y_{1..j-1}$. We can thus compute the partition function matrices using standard dynamic programming.

$$\begin{aligned} Z^M_{i,j} &= (Z^M_{i-1,j-1} + Z^E_{i-1.j-1} + Z^F_{i-1,j-1}) e^{s(x_i, y_j)/T} \\ Z^E_{i,j} &= Z^M_{i,j-1} e^{g/T} + Z^E_{i.j-1} e^{ext/T} \\ Z^F_{i,j} &= Z^M_{i-1,j} e^{g/T} + Z^F_{i-1.j} e^{ext/T} \\ Z_{i,j} &= Z^M_{i,j} + Z^E_{i,j} + Z^F_{i,j} \end{aligned} \qquad (8)$$

Here $s(x,y)$ represents the score of aligning residue x_i with y_j, g is the gap open penalty, and *ext* is the gap extension penalty. The matrix $Z^M{}_{ij}$ represents the partition function of all alignments ending in x_i paired with y_j. Similarly $Z^E{}_{ij}$ represents the partition function of all alignments in which y_j is aligned to a gap and $Z^F{}_{ij}$ all alignments in which x_i is aligned to a gap. Boundary conditions and further details can be obtained from Miyazawa 1995.

Once the partition function is constructed, the posterior probability of x_i aligned to y_j can be computed as

$$P(x_i \sim y_j) = \frac{Z^M_{i-1,j-1} Z'^M_{i+1,j+1}}{Z} e^{s(x_i, y_j)/T} \qquad (9)$$

where $Z'^M{}_{i,j}$ is the partition function of alignments of subsequences $x_{i..m}$ and $y_{j..n}$ beginning with x_i paired with y_j and m and n are lengths of x and y respectively. This can be computed using standard backward recursion formulas as described in Durbin *et al.*, 1998.

In equation (9) $Z^M{}_{i-1,j-1}/Z$ and $Z'^M{}_{i+1,j+1}/Z$ represent the probabilities of all feasible suboptimal alignments (determined by the T parameter) of $x_{1..i-1}$ and $y_{1..j-1}$, and $x_{i+1.m}$ and $y_{j+1..n}$ respectively, where m and n are lengths of x and y respectively. Thus, equation (9) weighs alignments according to their partition function probabilities and estimates $P(x_i \sim y_j)$ as the sum of probabilities of all alignments where x_i is paired with y_j.

Probalign: Maximal expected accuracy alignment using partition function posterior probabilities

Recall the maximum expected accuracy alignment formulation described earlier. In order to compute such an alignment we need an estimate of the posterior probabilities. In this report, we utilize the partition function posterior probability estimates for constructing multiple alignments. For each sequence x, y in the input, we compute the posterior probability matrix $P(x_i \sim y_j)$ using equation (9). These probabilities are subsequently used to compute a maximal expected multiple sequence alignment using the Probcons methodology. First, the probabilistic consistency transformation (described in detail in Do *et al.*, 2005) is applied to improve the estimate of the probabilities. Briefly, the probabilistic consistency transformation is to re-estimate the posterior probabilities based upon three-sequence alignments instead of pairwise. Note that this does not mean alignments are recomputed; our estimation (as done in Probcons) is still fundamentally based upon pairwise alignments. It is possible to compute a partition function of three-sequence alignments, and subsequently estimate posterior probabilities directly from them. However, in this proof of concept study, we examine performance on pairwise alignments only.

After the probabilistic consistency transformation, sequence profiles are next aligned in a post-order walk along a UPGMA guide-tree. As is commonly done, UPGMA guide trees are computed using pairwise expected accuracy alignment scores. Finally, iterative refinement is performed to improve the alignment. This standard alignment procedure is described in more detail in Do *et al.*, 2005 and is implemented in the Probcons package (by the same authors).

We implement the Probalign approach by modifying the underlying Probcons program to read in arbitrary posterior probabilities for each pair of sequences in the input. All use of HMMs in the modified Probcons code is disabled. We modified the probA program of Muckstein *et al.*, 2002 for computing partition function posterior probability estimates. The Probalign program is represented algorithmically in Figure 1. Our current implementation is a beta version and mainly for proof of concept; however, the open source code is fully functional and is available with full support from http://www.cs.njit.edu/usman/probalign.

```
Probalign algorithm:
    1. For each pair of sequences $(x,y)$ in the input set
        a. Compute partition function matrices $Z(T)$
        b. Estimate posterior probability matrix $P(x_i \sim y_j)$
           for $(x,y)$ using equation (9)
    2. Perform the probabilistic consistency transformation and
       compute a maximal expected accuracy multiple alignment:
       align sequence profiles along a guide-tree and follow by it-
       erative refinement (Do et. al.).
```

Fig. 1. Probalign algorithmic description.

Experimental design

Alignment benchmarks. To test the accuracy of our method, we use three popular multiple protein sequence alignment benchmarks in the literature: BAliBASE, HOMSTRAD, and OXBENCH. BAliBASE (Thompson *et al.*, 2005) is the most widely used benchmark for assessing protein multiple sequence alignments. Each alignment is well curated and contains core regions that represent reliable structurally alignable portions of the alignment. These alignable regions are used for evaluating accuracy and the remainder is ignored. BAliBASE 3.0 contains 5 sets of multiple protein alignments, each with different characteristics. RV11 contains 38 equidistant families with sequence identity less than 20%, while RV12 contains 44 equidistant families with sequence identity between 20% and 40%. Both of these lack sequences with large internal insertions (> 35 residues). RV20 contains 41 families with > 40% similarity and an orphan sequence which shares less than 20% similarity with the rest of the family. RV30 contains 30 families which contain sub-families with > 40% similarity but < 20% similarity across the sub-families. RV40 contains sequences with large N/C terminal extensions and is the largest set with 49 alignments, while RV50 contains sequences with large internal insertions and is the smallest with 16 alignments. Both RV40 and RV50 contain sequences that share > 20% similarity with at least one other sequence in the set. Overall, there are 217 benchmark alignments within BAliBASE 3.0.

HOMSTRAD (Mizuguchi *et. al.*, 1998) is a curated database of structure-based alignments for homologous protein families. We use the April 2006 release for this study which contains 1033 families. HOMSTRAD contains all known protein structure clustered into homologous families and aligned on the basis of their 3-D structures.

OXBENCH (Raghava *et. al.*, 2003) is a set of structure-based alignments based on protein domains. It contains three sets of unaligned sequences: master, which are the unaligned protein domains in the true alignments; full, which contains full length unaligned proteins; and extended which contains additional proteins similar to the ones in unaligned master set. There are a total of 672 true master and extended alignments and 605 full sequence ones. Due to running time considerations, we exclude all datasets above 100 sequences.

Determining prediction accuracy. Given a true and estimated multiple sequence alignment, the accuracy of the estimated alignment is usually computed using two measures: the sum-of-pairs (SP) and the true column (TC) scores (Thompson *et al.*, 1999). SP is a measure of the number of correctly aligned residue pairs divided by the number of aligned residue pairs in the true alignment. TC is the number of correctly aligned columns divided by the number of columns in the true alignment. Both are standard measures of computing alignment accuracy.

Statistical significance. Statistically significant performance differences between the various alignment methods are calculated using the Friedman rank test (Kanji 1999), which is a standard measure used for discriminating alignments in benchmarking studies (Thompson *et. al*., 1999; Do *et al.*, 2005; Edgar 2004; Katoh *et al.*, 2005). Roughly speaking, the lower the reported P-value the less likely it is that the difference in ranking between the methods is due to chance. We consider P-values below 0.05 (a standard cutoff in statistics) to be statistically significant.

Programs compared and parameter settings. We compare Probalign to Probcons v1.1, MAFFT v5.851, and MUSCLE v3.6. These versions are the most current at the time of writing of this paper. We use the L-INS-i strategy of MAFFT, which is the most accurate according to latest benchmark tests by the MAFFT authors. The programs are compared using the scoring matrices and gap penalties recommended for their respective algorithms.

Probalign has two sets of parameters, one for the component that computes the posterior probabilities and the other for computing the maximal expected accuracy alignment. For the first component we use the Gonnet 160 scoring matrix (Gonnet *et. al.*, 1992) with gap open and gap extension penalties set to -22 and -1 respectively. The default value of T (thermodynamic temperature) was set to 5 after comparing values 1 through 9 on BAliBASE RV11 (see Table 1). For the second component, we use the exact same default parameters as that of Probcons, i.e. two rounds of probabilistic consistency and at most 100 rounds of iterative refinement.

3 RESULTS

3.1 Effect of thermodynamic temperature

We first look at the effect of different values of the thermodynamic temperature T on Probalign. Table 1 shows that T=5 is optimal on RV11. These settings of T appear to work well for the Gonnet 160 matrix and its affine gap penalties; therefore, we set T=5 for the remainder of our experiments.

Table 1. Effect of different thermodynamic temperatures on Probalign on RV11 subset of BAliBASE 3.0.

T	Mean SP / TC	T	Mean SP / TC	T	Mean SP / TC
1	51.43 / 24.89	4	65.23 / 43.03	7	60.28 / 36.58
2	55.06 / 29.08	5	**69.32 / 45.26**	8	49.51 / 25.76
3	57.90 / 32.39	6	66.18 / 40.87	9	41.12 / 18.84

3.2 Benchmark comparisons

In Table 2 we compare mean SP scores and TC of Probalign to other methods on BAliBASE 3.0. Probalign averages are the highest on the RV11, RV12, and RV40 subsets, as well as the full

BAliBASE dataset. MAFFT does better on the remaining three datasets. Although the differences are small, Probalign ranks statistically significantly higher than all three methods on RV12, RV40, and the full BAliBASE dataset (see Table 3). No method ranked statistically significantly higher than Probalign on any of the BAliBASE subsets.

Table 2. Mean SP / TC scores on BAliBASE 3.0.

Data	Probalign	MAFFT	Probcons	MUSCLE
RV11	**69.3 / 45.3**	67.1 / 44.6	67.0 / 41.7	59.3 / 35.9
RV12	**94.6 / 86.2**	93.6 / 83.8	94.1 / 85.5	91.7 / 80.4
RV20	92.6 / 43.9	**92.7 / 45.3**	91.7 / 40.6	89.2 / 35.1
RV30	85.2 / 56.4	**85.6 / 56.9**	84.5 / 54.4	80.3 / 38.3
RV40	**92.2 / 60.3**	92.0 / 59.7	90.3 / 53.2	86.7 / 47.1
RV50	89.3 / 55.2	**90.0** / 56.2	89.4 / **57.3**	85.7 / 48.7
All	**87.6 / 58.9**	87.1 / 58.6	86.4 / 55.8	82.5 / 48.5

Table 3. P-values of Friedman rank test on BAliBASE TC scores. In all cases of statistical significance (< 0.05) Probalign is ranked higher. NS indicates non statistically significant.

Method	RV11	RV12	RV20	RV30	RV40	RV50	All
MAFFT	NS	< 0.005	NS	NS	< 0.005	NS	< 0.005
Probcons	0.049	0.0233	NS	NS	< 0.005	NS	< 0.005
MUSCLE	< 0.005	< 0.005	0.008	< 0.005	< 0.005	NS	< 0.005

We also test Probcons by retraining (on BAliBASE 3.0) with single and pair emission probabilities set to the background and mutation matrix probabilities of Gonnet 160. In this way we can test if the Probalign improvements are purely a result of scoring matrix differences. The performance of Probcons performance does not improve. In fact, it actually does worse than with training on the (default) Blosum 62 matrix.

Table 4 compares the CPU running time of Probalign to the other methods on RV11 and RV12 subsets of BAliBASE. While Probalign is the slowest, its running time is still tractable. Our current beta implementation is a pipeline of C++ programs and Perl scripts linked by system calls. An integrated version (which is in progress) will yield a much faster implementation.

Table 4. Mean CPU time (in seconds) on RV11 and RV12 subsets of BAliBASE 3.0.

Data	Probalign	MAFFT	Probcons	MUSCLE
RV11	6.64	0.98	3.65	0.71
RV12	17.73	1.28	10.46	0.74

Finally, Table 5 compares mean SP and TC scores on the HOMSTRAD and OXBENCH benchmarks. Probalign mean SP and TC scores rank highest on HOMSTRAD, OXBENCH, and OXBENCH-full with P-value < 0.005. Moreover, on the OXBENCH-extended dataset, no method ranked statistically significantly higher than Probalign. In fact, Probalign ranks higher than Probcons on OXBENCH-extended with P-value 0.014.

Table 5. Mean SP / TC scores on HOMSTRAD and OXBENCH.

Data	Probalign	MAFFT	Probcons	MUSCLE
HOMSTRAD	**82.2 / 77.9**	80.4 / 75.9	81.9 / 77.4	80.8 / 76.3
OXBENCH	**89.8 / 85.1**	88.4 / 83.2	89.3 / 84.2	89.4 / 84.4
OXBENCH (full)	**84.0 / 77.0**	82.8 / 75.3	83.2 / 75.7	82.6 / 74.8
OXBENCH (extend)	92.0 / 89.6	**92.5 / 90.0**	92.4 / 89.8	91.8 / 89.0

3.3 Simulation of N/C terminal extensions

Probalign's performance improvement is most significant over all methods on the RV40 subset of BAliBASE. Recall that this dataset contains sequences with long N/C terminal extensions. We rely on simulation to further test Probalign's improvement on this type of data. We begin by computing the maximum parsimony model trees (with edge lengths) on arbitrary selected alignments from the RV11 subset of BAliBASE 3.0. We select the BB11003, BB11004, BB11008, BB11009, and BB11010 alignments, all of which contain four sequences and branch length ranging from conservative to divergent. For each tree, we generate a root protein sequence with the same background probability distribution as Dayhoff's. We define core regions of this sequence as randomly selected contiguous region (with probability 0.25) ranging from length 1 to 30 (with uniform probability). We then evolve sequences using the ROSE model (Stoye *et. al.*, 1998). However, in the defined core regions, the mutation probability is reduced (by half) and no insertion deletions are allowed.

Briefly, ROSE interprets each branch length as PAM units of evolution. On a branch of length k, the probability of substitution is given by M^k where M is the PAM1 mutation probabilities. For insertion (or deletion) it randomly picks an amino acid with probability *insert_threshold* * *branch_length* * *sequence_length* and inserts (or deletes) a sequence of length given by an exponential distribution. Once the simulated sequences are generated, we attach a randomly generated sequence to each end of each sequence with probability 0.25, which constitute our artificial N/C extensions.

For each model tree, we produce a root sequence of length 100, and the (insertion, deletion) thresholds are set to (0.0005, 0.000125), meaning the deletion threshold is 1/4th the insertion. We generate 100 sequence sets for each model tree, and align using Probalign, MAFFT, and Probcons. The alignments are compared against the core regions of the true alignment (known by simulation). Table 6 shows that Probalign wins for all model trees. Probalign SP and TC scores also rank higher than all methods with P-value < 0.05 (except for BB11009 where all methods do equally well). We also examined performance on simulated data containing long internal insertions, along with the N/C extensions, and saw similar results (data not shown).

Table 6. Mean SP / TC scores on different model trees. Also shown are average branch lengths (PAM units of evolution) for each model tree.

Model tree	Probalign	MAFFT	Probcons
BB11003 (164)	**77.1 / 63.7**	72.7 / 58.2	72.4 / 56.9
BB11004 (132)	**89.5 / 83.0**	86.7 / 78.3	86.8 / 78.5
BB11008 (92)	**97.9 / 95.9**	96.8 / 93.9	96.5 / 93.3
BB11009 (33)	**99.8 / 99.7**	99.8 / 99.7	99.8 / 99.6
BB11010 (184)	**63.4 / 46.9**	58.1 / 41.0	60.1 / 414

3.4 Datasets with long and variable length sequences

Not only does the RV40 subset contain sequences with large N/C extension, but they are also highly variable in length. In fact, many constituent proteins are at least 1000 residues in length. Based on our results thus far, we conjecture that Probalign does best when presented such datasets. To test this hypothesis, we select all unaligned datasets in BAliBASE 3.0 where the standard deviation in sequence length is at least 100 or 200 and the maximum length is at least 500 or 1000. For these four possible permutations, we compare the mean SP and TC scores of Probalign to the other methods (Table 7).

Table 7. Mean SP / TC scores on BAliBASE 3.0 datasets with standard deviation of length at least 100 and 200 and maximum sequence length at least 500 and 1000.

Max length / Standard dev.	Probalign	MAFFT	Probcons	MUSCLE
500 / 100	**88.4** / 56.6	88.0 / **58.0**	86.7 / 51.6	81.5 / 42.5
500 / 200	**88.5 / 54.6**	87.0 / 51.9	87.2 / 48.9	81.9 / 42.4
1000 / 100	**91.4 / 58.1**	90.4 / 55.7	89.7 / 51.6	84.3 / 44.1
1000 / 200	**90.7 / 55.0**	89.3 / 51.4	89.2 / 48.7	83.2 / 42.5

Table 7 shows that the improvement of Probalign over other methods increases as both the standard deviation of the mean length and the maximum sequence length increases. The Probalign mean column score (TC) is 2.8%, 2.4%, and 3.7% better than MAFFT at the 500/200, 1000/100, and 1000/200 settings, respectively, and at least 5% better than Probcons on all four combinations. Furthermore, even though the mean TC is lower than that of MAFFT in row one, *Probalign ranked higher than all methods on each of the four settings with P-value < 0.005 (for both TC and SP scores)*.

Table 8 shows mean SP and TC scores broken down for each BAliBASE subset but containing only those datasets with maximum sequence length at least 1000 and standard deviation of length at least 100 and 200. We omit MUSCLE from this comparison since it is poorest on these types of datasets. At the 1000/100 setting, Probalign mean TC score is at least 2.8%, 3%, and 4% better than MAFFT and Probcons on RV12, RV30, and RV40 subsets, respectively. At the 1000/200 setting, TC improvement on both RV30 and RV40 increases to at least 5%. However, only on RV40 is Probalign statistically significantly ranked highest for both SP and TC score (with P-value < 0.005). No method ranked statistically significantly higher than Probalign.

Table 8. Mean SP / TC scores for datasets with max sequence length at least 1000 and standard deviation of length at least 100 and 200 for each BAliBASE subset. The number of datasets in each BAliBASE subset (RV11 through RV50) satisfying these criteria is indicated in parentheses.

Max length / Standard dev.	Probalign	MAFFT	Probcons
RV11 1000 / 100 (1)	**62.5 / 39.0**	55.2 / 36.0	62.8 / 38.0
1000 / 200 (1)	**62.5 / 39.0**	55.2 / 36.0	62.8 / 38.0
RV12 1000 / 100 (5)	**93.6 / 81.6**	91.5 / 77.0	92.3 / 78.8
1000 / 200 (5)	**93.6 / 81.6**	91.5 / 77.0	92.3 / 78.8
RV20 1000 / 100 (6)	**92.3 / 42.0**	91.7 / 41.0	91.0 / 38.5
1000 / 200 (5)	**91.6 / 34.6**	90.9 / 34.0	90.1 / 30.4
RV30 1000 / 100 (3)	**90.8 / 67.3**	90.6 / 64.3	89.4 / 63.3
1000 / 200 (1)	**77.2 / 40.0**	76.1 / 34.0	73.6 / 35.0
RV40 1000 / 100 (25)	**92.7 / 59.3**	91.0 / 54.8	89.9 / 48.2
1000 / 200 (20)	**93.0 / 57.3**	90.8 / 52.1	90.6 / 47.6
RV50 1000 / 100 (6)	88.1 / 48.5	**91.2 / 55.8**	89.7 / 52.2
1000 / 200 (4)	85.0 / 43.5	**89.1 / 45.8**	87.3 / 45.8

On RV50, MAFFT is the winner on both the full dataset (see Table 2) and on the subsets in Table 8, but not statistically significantly ranked higher. By reducing the gap extension penalty (to allow for large internal insertions), Probalign's TC score improves considerably (but not statistically significantly) as shown in Table 9 below. The TC score with 0.2 gap extension penalty is 3.2% better than Probcons and MAFFT at the 1000/200 setting.

Table 9. Mean SP / TC scores for the full RV50 BAliBASE dataset (long internal insertions) in row two and for RV50 datasets with long and heterogeneous length sequences (last two rows). The number of datasets meeting these criteria is indicated in parentheses.

RV50 Dataset	Probalign (gap ext 0.2)	Probalign (gap ext 1.0)	MAFFT	Probcons
Complete	87.8 / 56.4	89.3 / 55.2	**90.0** / 56.2	89.4 / **57.3**
Max len / Std dev				
1000/100 (6)	88.2 / **56.0**	88.1 / 48.5	**91.2 /** 55.8	89.7 / 52.2
1000/200 (4)	85.9 / **49.0**	85.0 / 43.5	**89.1 /** 45.8	87.3 / 45.8

We perform one more test here to examine performance on heterogeneous length sequences. We consider reference set 6 of BAliBASE 2.0 (Thompson *et al.*, 2001) containing repeats. Repeats are much smaller than the original sequence and most of the repeat datasets containing highly variable length sequences. Reference 6 of BAliBASE contains 13 reference alignments of repeats and several more repeat datasets classified into six different subsets. We refer the reader to Thompson *et al.*, 2001 for complete classification details. We gather all datasets in reference 6 (for a total of 77) and considered only those with maximum sequence length at least 500 and 1000, and standard deviation of length at least 100, 200, 300, and 400. Again, we omit MUSCLE because it performs worse than the three other methods on this type of data.

Table 10. Mean SP / TC scores on BAliBASE 2.0 reference 6 (repeat) datasets with std. deviation of length at least 100, 200, 300, and 400, and maximum sequence length at least 500 and 1000. Indicated in parentheses are the number of datasets meeting these conditions.

Max length / Standard dev.	Probalign	MAFFT	Probcons
500 / 100 (40)	**89.1 /** 44.9	87.3 / **49.0**	87.4 / 38.6
500 / 200 (21)	**88.3** / 43.8	85.0 / **46.4**	86.7 / 40.0
500 / 300 (9)	**95.3 / 61.0**	82.6 / 51.3	87.3 / 46.6
500 / 400 (5)	**94.6 / 55.0**	72.0 / 38.2	79.8 / 38.0
1000 / 100 (15)	**90.2 / 43.3**	82.4 / 36.9	85.4 / 27.6
1000 / 200 (12)	**89.2 / 38.2**	79.7 / 32.4	83.6 / 27.7
1000 / 300 (7)	**94.5 / 52.8**	78.3 / 42.4	83.9 / 34.6
1000 / 400 (5)	**94.6 / 55.0**	72.0 / 38.2	79.8 / 38.0

The Probalign improvements on these datasets are the largest observed so far (see Table 10 above). As the maximum sequence length and the standard deviation in length increases so does the Probalign improvement. When standard deviation of length is at least 300 and 400, Probalign SP and TC score is at least 10% and 15% better than the next best method. While no method is ranked statistically significantly better than any other on these datasets, these large Probalign improvements gained warrant significant merit.

4 DISCUSSION

Probalign's improved performance arises from consideration of suboptimal alignments. Let us look at equation (9) where the posterior probabilities are estimated. Here, $Z^{M}_{i-1,j-1}/Z$ and $Z'^{M}_{i+1,j+1}/Z$ represent the probabilities of all alignments of $x_{1..i-1}$ and $y_{1..j-1}$, and

$x_{i+1.m}$ and $y_{j+1..n}$ where *m* and *n* are lengths of *x* and *y* respectively. Strictly speaking, we are not looking at *all* alignments of $x_{1..i-1}$ and $y_{1..j-1}$ but only a subset of suboptimal alignments determined by the *T* parameter, which is analogous to the thermodynamic temperature. These suboptimal alignments may in fact be more biologically accurate, while not necessarily the most optimal under the employed scoring scheme. This result was reported previously (Muckstein *et al.*, 1998) when examining several thousand suboptimal pairwise alignments (generated using the partition function) for a particular pair of proteins. Many of the suboptimal alignments were deemed to be more biologically relevant than the optimal. This result is the underlying motivation for our combined Probalign approach.

Further insight into Probalign is gained by generating an ensemble of high probability suboptimal pairwise alignments using stochastic backtracking of the partition function matrix (as described in Muckstein *et al.*, 2002), and then estimating $P(x_i \sim y_j)$ as the fraction of alignments where x_i is paired with y_j. This method produces almost exactly the same results as when using equation (9). In light of this result, it is now perhaps easier to see why Probalign is particularly better than other methods at aligning heterogeneous datasets, which are long in length. In such datasets, regions that are highly similar will be preserved in most suboptimal alignments, even though they may not be perfectly aligned in the optimal one (which, as we have seen in our experiments, is usually the case).

The results in this study allow us to directly compare posterior probability estimates using the Probcons and Probalign techniques. Both follow the exact same strategy once the probabilities are specified. Probalign has the advantage over Probcons of not having to learn model parameters from training data. This important distinction makes Probalign applicable to situations where a diverse range of training data is not readily available (i.e., motif searching, repeat alignments, widely variable lengths, RNA and DNA sequences). On the other hand, the learning algorithm of Probcons can learn optimal gap parameters directly and not have to resort to hand-tuned ones the way that Probalign requires.

By generating a high probability alignment ensemble (for a given pair of sequences) it is possible to assign weights to different alignments based upon biological features. For example, future work could assign weights based on features such as number of gapless long hydrophobic regions or number of hydrophilic residues around gaps (similar to what is done in Do *et al.*, 2006). Alternative approaches for generating alignment ensembles remain to be explored. The applicability of Probalign for constructing accurate RNA alignments and also those that produce accurate phylogenetic trees also remains to be seen. Probalign's performance on long and heterogeneous length datasets suggests it may be useful in aligning and detecting motifs in long DNA genomic regions. Finally, other alignment programs based upon the Probcons framework may also perform better with the partition function posterior probabilities (Paten 2005; Schwartz *et al.*, 2006).

ACKNOWLEDGEMENTS

We thank Chuong Do and Robert Edgar for helpful discussions about Probcons and multiple sequence alignment techniques. We also thank anonymous referees for providing valuable feedback. Computational experiments were performed on the CIPRES cluster supported by NSF (EF0331654). DRL is supported, in part, by NIH R01 GM073082-0181.

REFERENCES

C. Notredame, (2002) Recent progresses in multiple sequence alignment: a survey, *Pharmacogenomics*,,3(1) pp:131-144.

D. La, B. Sutch, and D.R. Livesay, (2005) Predicting protein functional sites with phylogenetic motifs, *Proteins* 58 pp:309-320.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, (1998) Biological sequence analysis: probabilistic models of proteins and nucleic acids, Cambridge University Press

J. D. Thompson, D. G. Higgins, and T. J. Gibson, (1994) ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties, and weight matrix choice. *Nucleic Acids Research* 27(13) pp:2682-2690.

A. R. Subramanian, J. Weyer-Menkhoff, M. Kaufmann, and B. Morgenstern, (2005) Dialign-T: an improved algorithm for segment-based multiple sequence alignment, *BMC Bioinformatics* 6 pp:66.

C. Notredame, D. Higgins, and J. Heringa, (2000) T-Coffee: a novel method for multiple sequence alignments, *Journal of Molecular Biology* 302 pp:205-217.

C. B. Do, M. S. P. Mahabhashyam, M. Brudno, and S. Batzoglou, (2005) PROBCONS: probabilistic consistency based multiple sequence alignment. *Genome Research* 15 pp:330-340.

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

K. Katoh, K. Misawa, K. Kuma, and T. Miyata, (2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment, *Nucleic Acids Research* 33 pp:511-518.

S. Miyazawa, (1995) A reliable sequence alignment method based upon probabilities of residue correspondences, *Protein Engineering* 8(10) pp:999-1009.

J. D. Thompson, P. Koehl, R. Ripp, and O. Poch, (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark, *Proteins* 61 pp:127-136.

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

M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt, (1978) A model for evolutionary change in proteins, In M. O. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, 5 pp:345-352, National Biochemical Research Foundation, Washington DC

U. Muckstein, I. L. Hofacker, and P. F. Stadler, (2002) Stochastic pairwise alignments, *Bioinformatics* 18 Suppl 2 pp:S153-160.

G. P. S. Raghava, S. M. J. Searle, P. C. Audley, J. D. Barber, and G. J. Barton, (2003) OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy, *BMC Bioinformatics* 4:47.

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

G. K. Kanji, (1999) 100 Statistical Tests, *Sage Publications,*

C. B. Do, S. S. Gross, and S. Batzoglou, (2006) CONTRAlign: Discriminative Training for Protein Sequence Alignment, *Proceeding of Tenth Annual International Conference on Computational Molecular Biology(RECOMB)*

B. Paten, (2005) http://www.ebi.ac.uk/~bjp/pecan/

A. S. Schwartz, E. Myers, and L. Pachter, (2006) Alignment metric accuracy, *acrxiv.org/avs/q-bio.QM/0510052*

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

A. Bahr, J. D. Thompson, J. C. Thierry, and O. Poch, (2001) BAliBASE (Benchmark Alignment dataBASE) enhancements for repeats, transmembrane sequences, and circular permutations, *Nucleic Acids Research* 29(1) pp:323-326.

J. Stoye, D. Evers, and F. Meyer, (1998) Rose: generating sequence families, *Bioinformatics* 14(2) pp:157-163.

G. H. Gonnet, M. A. Cohen, and S. A. Brenner, (1992) Exhaustive matching of the entire protein sequence database, *Science,* 256(5062) pp:1443-1445

S. Karlin and S. F. Altschul, (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schmes, *Proceedings of National Academy of Sciences of USA,* 87(6) pp:2264-2268

S. F. Altschul, (1993) A protein alignment scoring system sensitive at all evolutionary distances, *Journal of Molecular Evolution,* 36(3) pp:290-300