BIOINFORMATICS ORIGINAL PAPER
Vol. 26 no. 16 2010, pages 1958–1964
doi:10.1093/bioinformatics/btq338

Sequence analysis

Advance Access publication June 23, 2010

MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities

Yongchao Liu*, Bertil Schmidt and Douglas L. Maskell

School of Computer Engineering, Nanyang Technological University, Singapore 639798

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Multiple sequence alignment is of central importance to bioinformatics and computational biology. Although a large number of algorithms for computing a multiple sequence alignment have been designed, the efficient computation of highly accurate multiple alignments is still a challenge.

Results: We present *MSAProbs*, a new and practical multiple alignment algorithm for protein sequences. The design of MSAProbs is based on a combination of pair hidden Markov models and partition functions to calculate posterior probabilities. Furthermore, two critical bioinformatics techniques, namely weighted probabilistic consistency transformation and weighted profile–profile alignment, are incorporated to improve alignment accuracy. Assessed using the popular benchmarks: BAliBASE, PREFAB, SABmark and OXBENCH, MSAProbs achieves statistically significant accuracy improvements over the existing top performing aligners, including ClustalW, MAFFT, MUSCLE, ProbCons and Probalign. Furthermore, MSAProbs is optimized for multi-core CPUs by employing a multi-threaded design, leading to a competitive execution time compared to other aligners.

Availability: The source code of MSAProbs, written in C++, is freely and publicly available from http://msaprobs.sourceforge.net.

Contact: liuy0039@ntu.edu.sg

Received on March 21, 2010; revised on May 22, 2010; accepted on June 18, 2010

1 INTRODUCTION

Multiple sequence alignment is of central importance to bioinformatics and computational biology. The approach for producing an optimal multiple sequence alignment is to simultaneously align multiple sequences using dynamic programming. Unfortunately, this approach is impractical for alignments of more than a few sequences, due to its high computational cost. Therefore, many heuristics have been proposed to compute nearly optimal alignments, such as progressive alignment (Feng and Doolittle, 1987), iterative alignment (Barton and Sternberg, 1987; Berger and Munson, 1991; Corpet, 1988; Subbiah and Harrison, 1989) and alignment based on profile hidden Markov models (Krogh *et al*., 1994; Rabiner, 1989). State-of-the-art multiple sequence alignment algorithms tend to share some common techniques to improve alignment accuracy, including combining iterative alignment with progressive alignment, and introducing consistency-based schemes. These algorithms are typically assessed on publicly available benchmark data sets, including: BAliBASE (Bahr, 2001; Thompson *et al*., 1999, 2005), PREFAB (Edgar, 2004a), SABmark (Van Walle *et al*., 2004) and OXBENCH (Raghava *et al*., 2003). Currently, the best performing multiple sequence alignment algorithms based on these benchmark tests are T-Coffee (Notredame *et al*., 2000), MAFFT (Katoh *et al*., 2002, 2005), MUSCLE (Edgar, 2004a, 2004b), ProbCons (Do *et al*., 2005) and Probalign (Roshan and Livesay, 2006).

ClustalW (Larkin *et al*., 2007; Thompson *et al*., 1994) is historically one of the most popular multiple sequence alignment programs (with more than 26 000 citations in the ISI Web of Science), complying with the typical progressive alignment pipeline. T-Coffee introduced a consistency-based objective function COFFEE (Notredame *et al*., 1998) to progressive alignment by employing a primary library generated from pairwise global and local alignments to form three-way alignments. MAFFT uses the fast Fourier transform method for rapid identification of homologous regions. It then iteratively refines alignment results after performing an initial progressive alignment. The accuracy of MAFFT is further improved by introducing a consistency approach incorporating pairwise information into the objective function. MUSCLE works by iteratively refining alignment results with progressive alignment at the core, adopting a log-expectation scoring scheme instead of the conventional weighted sum-of-pairs scoring systems. ProbCons employs maximum expected accuracy as an objective function, and introduces a probabilistic consistency approach, based on pair hidden Markov model (pair-HMM) posterior probabilities (Durbin *et al*., 1998), to form three-way alignments. Probalign adopts a very similar strategy to ProbCons, but employs a partition function (Miyazawa, 1995) to calculate posterior probabilities instead of using a pair-HMM.

In this article, we present MSAProbs, a new and practical multiple protein sequence alignment algorithm designed by combining a pair-HMM and a partition function to calculate posterior probabilities. We further investigate two critical bioinformatics techniques, namely weighted probabilistic consistency transformation and weighted profile–profile alignment, to achieve high alignment accuracy. In addition, MSAProbs is optimized for modern multi-core CPUs by employing a multi-threaded design in order to reduce execution time. Assessed on the four popular benchmarks: BAliBASE, PREFAB, SABmark and OXBENCH, MSAProbs demonstrates significant alignment accuracy improvements over several leading aligners: ClustalW, MAFFT, MUSCLE, ProbCons and Probalign, with competitive execution time. Since T-Coffee has been proven to be inferior to MAFFT, MUSCLE and ProbCons in these papers (Do *et al*., 2005; Edgar, 2004a; Katoh *et al*., 2005),

*To whom correspondence should be addressed.

we have decided not to include the comparison to it. ClustalW is compared because of its very fast speed (see the 'Results' section).

2 METHODS

MSAProbs can be classified as a progressive alignment approach to computing multiple protein sequence alignments. It works by (i) calculating all pairwise posterior probability matrices using both a pair-HMM and a partition function; (ii) calculating a pairwise distance matrix using the posterior probability matrices; (iii) constructing a guide tree from the pairwise distance matrix, and calculating sequence weights; (iv) performing a weighted probabilistic consistency transformation of all pairwise posterior probability matrices; and (v) computing a progressive alignment along the guide tree using the transformed posterior probability matrices. To further improve alignment accuracy, an additional iterative refinement is performed as a post-processing step of stage (v).

2.1 Posterior probability matrix computation

Given are two protein sequences x and y of a protein sequence dataset S. We define x_i to denote the i-th amino acid in x, and y_j to denote the j-th amino acid in y. Let A be the space of all possible global alignments of x and y. Let $a^* \in A$ be the '*true*' alignment of x and y. Following ProbCons, the posterior probability that x_i is aligned to y_j (denoted as $x_i \sim y_j$) in a^*, is defined as

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

for all $1 \le i \le |x|$ and $1 \le j \le |y|$. The indicator function $\mathbf{1}\{cond\}$ returns 1 if the condition *cond* is true and 0, otherwise. $P(a|x, y)$ represents the probability that a is the *true* alignment a^*. Thus, $P(x_i \sim y_j \in a^* | x, y)$, i.e. $P(x_i \sim y_j)$ for short, can be considered as the probability that x_i is aligned to y_j in the true alignment a^*. The posterior probability matrix P_{xy} of x and y is a 2D table of size $|x| \times |y|$, consisting of the values $P(x_i \sim y_j)$ for $1 \le i \le |x|$ and $1 \le j \le |y|$. In MSAProbs, each pairwise posterior probability matrix is calculated by combining the probability matrices generated by a pair-HMM and a partition function as follows.

A pair-HMM calculates the pairwise probability matrix P^a_{xy} using the Forward and Backward algorithms, as described in Durbin *et al*. (1998). The partition function of alignments calculates the pairwise probability matrix P^b_{xy} through generating suboptimal alignments using dynamic programming. For all global alignments of x and y ending at position (i, j), we define $Z(i, j)$ to denote the partition function, $Z_M(i, j)$ to denote the partition function with x_i aligned to y_j, $Z_E(i, j)$ to denote the partition function with y_j aligned to a gap, and $Z_F(i, j)$ to denote the partition function with x_i aligned to a gap. The partition function can then be defined recursively as

$$\begin{aligned} Z_M(i,j) &= Z(i-1,j-1) e^{\beta sbt(x_i, y_j)} \\ Z_E(i,j) &= Z_M(i,j-1) e^{\beta\rho} + Z_E(i,j-1) e^{\beta\sigma} \\ Z_F(i,j) &= Z_M(i-1,j) e^{\beta\rho} + Z_F(i-1,j) e^{\beta\sigma} \\ Z(i,j) &= Z_M(i,j) + Z_E(i,j) + Z_F(i,j) \end{aligned} \quad (2)$$

where *sbt* is the substitution matrix, ρ ($\rho \le 0$) is the gap open penalty, σ ($\sigma \le 0$) is the gap extension penalty, and β is a parameter measuring the deviation between suboptimal and optimal alignments. A substitution matrix *sbt* gives the substitution rates of amino acids in proteins, derived from alignments of protein sequences. The boundary conditions and more details can be obtained from Miyazawa (1995). Using this partition function, $P(x_i \sim y_j)$ is defined as

$$P(x_i \sim y_j) = \frac{Z_M(i-1,j-1) Z'_M(i+1,j+1)}{Z} e^{\beta sbt(x_i, y_j)} \quad (3)$$

where $Z'_M(i, j)$ represents the partition function of all the reverse alignments starting from position $(|x|, |y|)$ and ending at (i, j) with x_i aligned to y_j, for $1 \le i \le |x|$ and $1 \le j \le |y|$.

After computing the probability matrix P^a_{xy} using pair-HMM and P^b_{xy} using partition function, the final probability matrix P_{xy} is calculated by combining these two matrices as the root mean square of the corresponding values in P^a_{xy} and P^b_{xy}.

$$P_{xy}(x_i \sim y_j) = \sqrt{\frac{P^a_{xy}(x_i \sim y_j)^2 + P^b_{xy}(x_i \sim y_j)^2}{2}} \quad (4)$$

The underlying motivation of combining the pair-HMM and partition function probabilistic models for posterior probabilities calculation is inspired by the alignment accuracy of sequences with long N/C-terminal extensions in BAliBASE benchmark, reported in the papers Do *et al*. (2005) and Roshan and Livesay (2006). In Do *et al*. (2005), the authors argue that the alignment accuracy of sequences with long N/C-terminal extensions, where local alignments tend to be more successful, might be improved by incorporating a local alignment probabilistic model. Moreover, in Roshan and Livesay (2006), the partition function shows superior performance on this type of data sets, indicating that the partition function probabilistic model might be more successful in locating highly similar regions. These two points have inspired the approach taken in this article, i.e. the combination of the two probabilistic models, to multiple sequence alignment.

2.2 Pairwise distance computation

After obtaining the probability matrix P_{xy} for each x, $y \in S$, a pairwise global alignment is performed to obtain the optimal global alignment score $GScore(x, y)$, where all match/mismatch scores are given by P_{xy} and gap penalties are set to zero. The optimal global alignment score $S(i, j)$ ending at position (i, j) of x and y, for $1 \le i \le |x|$ and $1 \le j \le |y|$, is recursively defined as

$$S(i,j) = \max \begin{cases} S(i-1,j-1) + P_{xy}(x_i \sim y_j) \\ S(i-1,j) \\ S(i,j-1) \end{cases} \quad (5)$$

where S(|x|,|y|) stores the final optimal global alignment score $GScore(x,y)$.

Many algorithms approximate pairwise distances from fractional identities in optimal global or local alignments obtained using a traceback procedure. In this work, we exploit an alternative approximation that calculates the pairwise distance $d(x,y)$ from $GScore(x, y)$ using Equation (6), defined as

$$d(x,y) = 1 - \frac{GScore(x,y)}{\min\{|x|, |y|\}} \quad (6)$$

This approximation is inspired by the fact that for a sequence pair, many optimal global alignments, giving the same optimal global alignment score, might be obtained using the traceback procedure. In this case, when using fractional identities to approximate the pairwise distances, the final distances are highly dependent on which optimal global alignments are chosen, because the fractional identities of these alignments generally are not identical. Our method avoids this dependence by using the optimal global alignment score.

2.3 Guide tree construction and sequence weighting

Given a pairwise distance matrix, a guide tree can be constructed using clustering methods such as neighbor-joining (Saitou and Nei, 1987; Studier and Keppler, 1988), UPGMA or its variants (Sneath and Sokal, 1973). MSAProbs implements the UPGMA that uses the linear combinatorial strategy to construct the guide tree, as described in Sneath and Sokal (1973). For this guide tree, the distance between the new cluster z, formed by merging two existing clusters x and y, and a third cluster w (excluding x and y) is defined as

$$d(w,z) = \frac{d(w,x) \times Leafs(x) + d(w,y) \times Leafs(y)}{Leafs(x) + Leafs(y)} \quad (7)$$

where $Leafs(x)$ represents the number of leafs in cluster x.

Sequence weighting is further considered to be able to correct for unequal sampling from a family of related proteins. After having constructed the guide tree, sequences are weighted following the tree topology. Among the available weighting schemes (Altschul *et al*., 1997; Gerstein *et al*. 1994; Gotoh, 1995; Henikoff and Henikoff, 1994; Thompson *et al*., 1994), we use the CLUSTALW (Thompson *et al*., 1994) method.

2.4 Weighted probabilistic consistency transformation

A probabilistic consistency transformation is used to re-estimate more accurate posterior probabilities of each sequence pair x and y by introducing another sequence z. Instead of re-computing the posterior probabilities based on three-sequence alignments, the transformation is performed based on the already computed probability matrices estimated from pairwise alignments. ProbCons uses the following heuristic to compute an approximate probabilistic consistency transformation,

$$P'_{xy} = \frac{1}{|S|}\left(2P_{xy} + \sum_{z \in S, z \neq x,y} P_{xz}P_{zy}\right) \quad (8)$$

where P'_{xy} is the new transformed posterior probability matrix of x and y, and $|S|$ is the number of sequences in S.

A drawback of the ProbCons approach is that it considers each sequence with identical significance. To avoid a biased sampling of sequences, we therefore derive a weighed probabilistic consistency transformation approach as follows. We define w_x to denote the weight of sequence x computed in the previous stage, and wN to denote the weighted number of sequences in S, i.e. the sum of sequence weights in S. This weighted approach is then defined as

$$P'_{xy} = \frac{1}{wN}\left((w_x + w_y)P_{xy} + \sum_{z \in S, z \neq x,y} w_z P_{xz}P_{zy}\right) \quad (9)$$

This motivation of the weighted approach is to obtain more accurate alignments than the non-weighted one. The transformations are further performed for a fixed number of iterations to refine the probabilities. In MSAProbs, two iterations (the default value) are used. This default value offers a good trade-off between alignment accuracy and execution time.

2.5 Progressive alignment

The final progressive alignment first aligns closely related sequences, and then distantly related sequences along the guide tree. Unlike ProbCons and Probalign (which are using an un-weighted profile–profile alignment model), MSAProbs uses a weighted one, which uses the sequence weights calculated in subsection 2.3. To compute a profile–profile alignment, the posterior probability matrix of the two profiles is calculated from the probability matrices of all sequence pairs x and y, where x and y are from different profiles respectively. After obtaining this probability matrix, the profile–profile alignment is carried out using Equation (5), where the match/mismatch scores are given by the probability matrix of the two profiles, and gap penalties are set to zero.

As a post-processing step, a randomized iterative alignment is employed to further improve alignment accuracy. This refinement randomly partitions S into two non-overlapped subsets, and then performs a profile–profile alignment of the two subsets. MSAProbs designs its own pseudo random number generator based on the linear congruential method for the random partition of S. The iterative refinement is designed to complete after a fixed number of iterations (10 iterations, by default).

2.6 Speed optimizations

The most time-consuming parts of MSAProbs are the posterior probability matrix computation, with a time complexity of $O(N^2L^2)$, and the weighted probabilistic consistency transformation, with a time complexity of $O(N^2L^3)$, where N is the number of sequences and L is the average sequence length. Because posterior probability matrices tend to be sparse with most entries near zero, the execution time of the probabilistic consistency transformation can be effectively reduced by using sparse matrix multiplication after transforming the matrices into sparse matrices (Do *et al*., 2005). However, this stage still has high time complexity. Our optimizations are focused on these two stages.

One optimization is to remove exponential computations in the recursive partition function equation. For a specific run, the parameters, including scoring matrix, gap penalties and β, are invariable. Hence, it is viable to pre-compute the exponential values in Equation (2) before performing the partition function computation. This leads to a significant decrease of execution time compared to directly computing using Equation (2). As multi-core CPUs have been commonplace, single-thread programs will result in the waste of compute resources of multi-core CPUs. In this case, our algorithm is optimized for multi-core CPUs by employing a multi-threaded design based on OpenMP (OpenMP, 2010), a compiler-directive-based application program interface (API) for explicitly directing multi-threaded, shared-memory parallelism. For the two stages, due to their irregular parallel natures, the DYNAMIC schedule policy of OpenMP is used to dynamically assign work to a team of parallel threads. For the posterior probability matrix computation stage, the matrix computation of a sequence pair is assigned to a thread, and for the probabilistic consistency transformation stage, the transformation for a sequence pair is assigned to a thread.

3 RESULTS

3.1 Accuracy measurement

To assess and rank different multiple protein sequence alignment algorithms, four benchmark data sets are used: BAliBASE, PREFAB, SABmark and OXBENCH. All tests are carried out on a PC with an Intel *i*7 quad-core 2.67 GHz processor and 12 GB RAM running the Linux operating system.

BAliBASE is the most widely used benchmark for assessing multiple protein sequence alignment algorithms. Each alignment is constructed by a combination of structure and sequence methods with manual refinement, and contains *core blocks*, regions for which reliable alignments are known to exist. BAliBASE 3.0 contains 386 reference alignments, which are organized into five reference sets. Reference 1 consists of equal-distant sequences, which are further organized into RV11 and RV12 reference subsets. RV11 consists of very distant sequences with <20% identity and RV12 consists of medium to divergent sequences with identities from 20% to 40%. Reference 2 (RV20) contains families with >40% identity and a highly divergent orphan sequence that shares <20% identity with the rest of the family. Reference 3 (RV30) consists of families that contains sub-families with >40% identity and <20% identity across sub-families. Reference 4 (RV40) consists of sequences with large N/C-terminal extensions, and Reference 5 (RV50) consists of sequences with large internal insertions. Accuracy evaluation on BAliBASE 3.0 is only scored with respect to core blocks.

PREFAB 4.0 is a fully automatically generated benchmark containing 1681 reference alignments. Each pair of sequences is supplemented with some homologous sequences found through PSI-BLAST (Altschul *et al*., 1997). Accuracy is assessed with respect to the pairwise structural alignments of the original two protein sequences using the consensus of FSSP (Holm and Sander, 1998) and CE (Shindyalo and Bourne, 1998) alignments. Since the pairwise structural alignments only cover some regions of the sequences, they can be treated as BAliBASE core blocks.

SABmark is also an automatically generated benchmark containing two sets of consensus regions based on SOFI (Boutonnet *et al*., 1995) and CF structural alignments of sequences selected from the ASTRAL (Brenner *et al*., 2000) database. This benchmark is divided into two subsets: *Twilight zone* and *Superfamilies*. Edgar (2010) argues that the pairwise reference alignments in SABmark are not generally consistent with a multiple alignment. It is therefore suggested to construct multiple alignments only from a consistent

subset of SABmark columns. Hence, SABRE (R.C. Edgar, personal communication), a subset of SABmark 1.65, is constructed by identifying mutually consistent columns (MCCs) in the pairwise reference structure alignment. SABRE contains 423 out of 634 SABmark groups by discarding groups having less than eight MCCs. MCCs can be considered analogous to BAliBASE core blocks for accuracy measurement. In this article, we use SABRE, instead of the original SABmark benchmark, to measure aligners.

OXBENCH is a set of structure based alignments generated by STAMP (Russell and Barton, 1992) from structures in the 3Dee database (Siddiqui *et al*., 2001). Accuracy measurement on OXBENCH can be conducted based on conservative columns, i.e. structurally conserved regions, which can also be considered analogous to BAliBASE core blocks.

In this article, alignments are scored according to *sum-of-pairs score* (SPS) and *column score* (CS) for BAliBASE, SABmark, and OXBENCH. SPS is defined as the number of correctly aligned residue pairs found in the test alignment divided by the total number of aligned residue pairs in core blocks of the reference alignment. CS is defined as the number of correctly aligned columns found in the test alignment divided by the total number of aligned columns in core blocks of the reference alignment. For PREFAB, alignments are scored on the reference structure pair using the quality score Q (Edgar, 2004a), which is equivalent to SPS. Statistical significance of the score differences between aligner pairs is calculated using the Wilcoxon matched-pair signed-rank test (Wilcoxon, 1947) with a P-value cutoff of 0.05. A collection of the above benchmarks is available at http://www.drive5.com/bench (R.C. Edgar, personal communication), and all the scores are calculated using the QSCORE scoring software (http://www.drive5.com/qscore), written by Robert C. Edgar.

MSAProbs has two sets of parameters: one for the pair-HMM and the other for the partition function. For the pair-HMM, MSAProbs uses the same emission probabilities and transition parameters as ProbCons (Do *et al*., 2005). For the partition function, MSAProbs uses the same parameters as Probalign, i.e. Gonnet 160 substitution matrix (Gonnet *et al*., 1992), a gap open penalty of -22, a gap extension penalty of -1 (not penalizing end gaps) and $\beta = 0.2$. These parameters are used by default.

3.2 Accuracy comparison to other algorithms

To assess the performance of MSAProbs for multiple protein sequence alignment, the above benchmarks are employed to compare MSAProbs with five top performing multiple sequence alignment algorithms: ClustalW version 2.0.12, MAFFT version 6.717, MUSCLE version 3.8.31, ProbCons version 1.12 and Probalign version 1.3. For MAFFT, the L-INS-i strategy, which yields the most accurate results among all the strategies of MAFFT, is used with the maximum iterative refinement (*–maxiterate* option) set to 1000. All the other algorithms (including MSAProbs) use their default parameters. All the scores in the following tables are multiplied by 100, and the best scores in each column are shown in bold.

On BAliBASE, Tables 1–3 show the mean SPS and CS scores of the six subsets and the overall dataset. MSAProbs achieves the highest SPS and CS scores on the overall BAliBASE data set, as well as all the subsets except for the RV40 subset (MAFFT produces the highest SPS score and Probalign gives the highest CS score on RV40). Table 3 also shows the overall runtime of each aligner. Table 4 shows the statistical significance of the score differences for all aligner pairs. From the table, both SPS and CS scores are best to distinguish between aligners, statistically ranking MSAProbs as the best.

On PREFAB, Tables 5 and 6 show the overall mean Q scores of all aligners and the statistical significance of the score differences for

Table 1. Mean SPS scores on BAliBASE 3.0 subsets

Aligner	RV11	RV12	RV20	RV30	RV40	RV50
MSAProbs	**74.63**	**94.86**	**94.35**	**88.20**	92.32	**90.90**
MUSCLE	65.75	92.32	91.50	84.23	86.31	85.28
MAFFT	69.18	93.68	93.62	87.81	**92.53**	90.14
Probalign	71.27	94.65	93.54	86.45	92.21	89.12
ProbCons	74.00	94.59	93.70	87.54	90.03	90.15
ClustalW	58.16	88.36	88.79	77.14	78.94	76.91

Table 2. Mean CS scores on BAliBASE 3.0 subsets

Aligner	RV11	RV12	RV20	RV30	RV40	RV50
MSAProbs	**53.70**	**87.45**	**53.93**	**63.44**	61.04	**61.43**
MUSCLE	43.31	82.00	42.22	47.67	45.32	47.51
MAFFT	48.35	84.46	48.88	61.83	59.99	58.29
Probalign	48.57	86.77	46.69	59.72	**61.23**	54.36
ProbCons	52.76	86.82	50.80	60.05	53.61	59.52
ClustalW	32.53	75.58	33.86	38.17	39.82	36.50

Table 3. Overall mean SPS and CS scores and runtime on BAliBASE 3.0

Aligner	SPS	CS	Time (hh:mm:ss)
MSAProbs	**89.09**	**64.51**	1:12:56
MUSCLE	84.33	53.17	0:16:11
MAFFT	87.50	61.07	0:41:05
Probalign	87.78	60.68	8:05:35
ProbCons	88.31	61.89	5:29:15
ClustalW	78.65	44.75	0:18:56

Table 4. Statistical significance of aligners on BAliBASE 3.0

Aligner	MSAProbs	MUSCLE	MAFFT	Probalign	ProbCons	ClustalW
MSAProbs		$<10^{-10}$	$<10^{-10}$	$<10^{-10}$	6.5×10^{-9}	$<10^{-10}$
MUSCLE	$<10^{-10}$*		$<10^{-10}$*	$<10^{-10}$*	$<10^{-10}$*	$<10^{-10}$
MAFFT	$<10^{-10}$*	$<10^{-10}$		0.02*	2.5×10^{-3}*	$<10^{-10}$
Probalign	$<10^{-10}$*	$<10^{-10}$	(0.10)		(0.88)	$<10^{-10}$
ProbCons	1.9×10^{-8}*	$<10^{-10}$	1.0×10^{-3}	(0.17)		$<10^{-10}$
ClustalW	$<10^{-10}$*	$<10^{-10}$*	$<10^{-10}$*	$<10^{-10}$*	$<10^{-10}$*	

Entries show P-value indicating the statistical significance of the mean scores differences between aligner pairs as measured using Wilcoxon matched-pair signed-rank test. The upper-right corner shows P-values calculated using SPS scores, and the lower-left corner shows P-values calculated using CS scores. * indicates the aligner on the left gives the worse performance, and the better performance, otherwise. For $P > 0.05$, the difference is considered insignificant and the P-value is shown in parentheses.

Table 5. Overall mean *Q* scores and runtime on PREFAB 4.0

Aligner	Q	Time (hh:mm:ss)
MSAProbs	**70.43**	03:34:36
MUSCLE	64.96	00:35:49
MAFFT	68.93	01:18:22
Probalign	68.72	23:59:22
ProbCons	68.43	15:32:43
ClustalW	59.33	01:01:12

Table 6. Statistical significance of aligners on PREFAB 4.0

Aligner	MSAProbs	MUSCLE	MAFFT	Probalign	ProbCons	ClustalW
MSAProbs		$<10^{-10}$	$<10^{-10}$	$<10^{-10}$	$<10^{-10}$	$<10^{-10}$
MUSCLE			$<10^{-10*}$	$<10^{-10*}$	$<10^{-10*}$	$<10^{-10}$
MAFFT				(0.78)	(0.07)	$<10^{-10}$
Probalign					3.1×10^{-3}	$<10^{-10}$
ProbCons						$<10^{-10}$
ClustalW						

Details are same as in Table 4.

Table 7. Mean SPS and CS scores and runtime on SABmark 1.65

Aligner	Twilight zone		Superfamilies		Overall		Time (mm:ss)
	SPS	CS	SPS	CS	SPS	CS	
MSAProbs	**43.10**	**23.04**	**66.48**	**46.20**	**60.51**	**40.29**	1:10
MUSCLE	35.94	17.29	61.46	40.61	54.95	34.66	0:48
MAFFT	39.19	18.85	63.33	42.55	57.17	36.50	1:14
Probalign	42.31	21.08	65.96	44.91	59.92	39.01	3:48
ProbCons	42.31	22.06	65.76	45.29	59.77	39.36	2:44
ClustalW	33.24	16.42	58.68	36.22	52.18	31.17	0:18

Table 8. Statistical significance of aligners on SABmark 1.65

Aligner	MSAProbs	MUSCLE	MAFFT	Probalign	ProbCons	ClustalW
MSAProbs		$<10^{-10}$	$<10^{-10}$	0.03	2.8×10^{-3}	$<10^{-10}$
MUSCLE	$<10^{-10*}$		$6.0\times10^{-5*}$	$<10^{-10*}$	$<10^{-10*}$	1.2×10^{-4}
MAFFT	$<10^{-10*}$	0.01		$<10^{-10*}$	$<10^{-10*}$	5.5×10^{-10}
Probalign	0.02*	$<10^{-10}$	1.3×10^{-9}		(0.57)	$<10^{-10}$
ProbCons	0.02*	$<10^{-10}$	$<10^{-10}$	(0.95)		$<10^{-10}$
ClustalW	$<10^{-10*}$	$6.6\times10^{-5*}$	$2.0\times10^{-7*}$	$<10^{-10*}$	$<10^{-10*}$	

Details are same as in Table 4.

all aligner pairs, respectively. From the tables, MSAProbs achieves statistically significant accuracy improvement over all the other aligners.

On SABmark, MSAProbs achieves the highest mean SPS and CS scores on the overall data set, as well as for the *Twilight zone* and *Superfamilies* subsets, as shown in Table 7. Meanwhile, MSAProbs statistically outperforms all the other aligners for both scores (Table 8).

Table 9. Overall mean SPS and CS scores and runtime on OXBENCH

Aligner	Overall		Time (mm:ss)
	SPS	CS	
MSAProbs	**90.06**	**81.70**	1:42
MUSCLE	89.50	80.67	0:23
MAFFT	88.86	79.48	1:02
Probalign	89.97	81.68	7:28
ProbCons	89.68	80.88	5:04
ClustalW	89.45	80.19	0:26

Table 10. Statistical significance of aligners on OXBENCH

Aligner	MSAProbs	MUSCLE	MAFFT	Probalign	ProbCons	ClustalW
MSAProbs		2.6×10^{-8}	$<10^{-10}$	(0.81)	$<10^{-10}$	3.7×10^{-4}
MUSCLE	$5.3\times10^{-7*}$		(0.07)	$6.3\times10^{-9*}$	$8.9\times10^{-3*}$	(0.84)
MAFFT	$<10^{-10*}$	0.01*		$<10^{-10*}$	$4.9\times10^{-6*}$	0.03*
Probalign	(0.42)	5.7×10^{-8}	$<10^{-10}$		5.4×10^{-6}	4.9×10^{-4}
ProbCons	$1.8\times10^{-10*}$	0.03	5.8×10^{-6}	$3.4\times10^{-7*}$		(0.27)
ClustalW	$3.2\times10^{-6*}$	(0.88)	4.7×10^{-2}	$1.4\times10^{-5*}$	(5.4×10^{-2})	

Details are same as in Table 4.

On OXBENCH, MSAProbs achieves the highest overall mean SPS and CS scores, as shown in Table 9. From the statistical perspective, the accuracy improvement of MSAProbs is statistically significant compared to MUSCLE, MAFFT, ProbCons and ClustalW, but has low significance compared to Probalign (Table 10). Nevertheless, MSAProbs yields the statistically highest SPS and CS scores on OXBENCH, even though its performance is indistinguishable from Probalign due to the lack of statistical significance.

While demonstrating dramatic improvement on alignment accuracy, MSAProbs still maintains competitive execution time (Tables 3, 5 and 9). On the two large benchmarks: BAliBASE and PREFAB, MSAProbs takes far shorter time than Probalign and ProbCons, even though it takes slightly longer time than MUSCLE, MAFFT and ClustalW. In particular, on PREFAB, ProbCons takes about 15.5 hours and Probalign takes about 24 h to complete the alignments, whereas MSAProbs only takes about 3.5 h on the same platform.

3.3 Comparison of MSAProbs variants

To understand how the various features of MSAProbs affect the alignment accuracy, some variants of MSAProbs are evaluated based on two algorithmic changes: (i) combining the pair-HMM and partition function posterior probabilities using weighted arithmetic mean, instead of root mean square; (ii) introducing un-weighted approaches for probabilistic consistency transformation and profile–profile alignment. The first algorithmic change is used for two purposes: one is to compare the performance difference between conventional arithmetic mean and root mean square for posterior probabilities calculation; and the other is to evaluate how the relative contributions of the two probabilistic models affect alignment accuracy. Using weighted arithmetic mean, the combined posterior

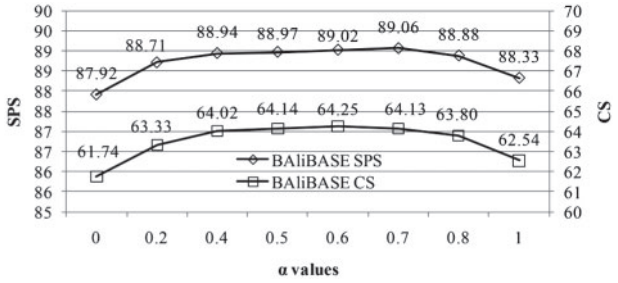

Fig. 1. Performance of the variants using representative α-values for weighted arithmetic mean calculation.

Table 11. Performance comparison of the variants using different weighting approaches

Scores	WPCT&WPPA	WPCT&PPA	PCT&WPPA	PCT&PPA
BAliBASE SPS	*89.09*	89.14	**89.28**	89.23
BAliBASE CS	**64.51**	64.47	**64.51**	*64.28*
PREFAB Q	**70.43**	70.08	69.72	*69.36*

WPCT (PCT) indicates the weighted (un-weighted) probabilistic consistency transformation, and WPPA (PPA) indicates the weighted (un-weighted) profile–profile alignment. The best scores in each row are shown in bold and the worst in italic.

probability is calculated as $P_{xy}(x_i \sim y_j) = \alpha \times P^a_{xy}(x_i \sim y_j) + (1-\alpha) \times P^b_{xy}(x_i \sim y_j)$, where $0 \leq \alpha \leq 1$. The relative contributions of the two probabilistic models can be changed by adjusting the value of α. In particular, only the pair-HMM posterior probabilities are used for $\alpha = 1$, and only the partition function posterior probabilities for $\alpha = 0$. The second algorithmic change is used to evaluate how weighting affects alignment accuracy.

We examined the performance of the variants that use weighted arithmetic mean on the BAliBASE 3.0 benchmark. In these tests, the SPS and CS scores of the resulting alignment are calculated for representative α values (Fig. 1). Figure 1 shows that both the SPS and CS scores increase from $\alpha = 0$, achieve the highest scores when α is around 0.6 and 0.7, and then decrease until $\alpha = 1$. This plot indicates that the single use of either probabilistic model is not able to give a strong increase in alignment accuracy. It further suggests that our combination of the two probabilistic models is a powerful approach for improving alignment accuracy. After comparing the scores in Fig. 1 and Table 3, it is obvious that the alignment accuracy using weighted arithmetic mean is inferior to that using root mean square. That is the underlying motivation of using root mean square instead of conventional arithmetic mean.

The effects of different weighting approaches on the alignment accuracy are examined on the BAliBASE 3.0 and PREFAB 4.0 benchmarks. We use root mean square for posterior probabilities calculation and keep all other conditions unchanged except for weighting approaches. The results of these tests are shown in Table 11. Define WPCT (PCT) to denote the weighted (un-weighted) probabilistic consistency transformation, and WPPA (PPA) to denote the weighted (un-weighted) profile–profile alignment. The four combinations of the weighting approaches lead to different alignment results on the two benchmarks (Table 11).

After comparing the scores of every combination, we can see that our weighted approaches do contribute to the whole accuracy improvement, but only by a small margin. From the table, it can be seen that the use of PCT and PPA (column 5) results in the lowest CS score for BAliBASE and the lowest Q score for PREFAB, and the use of WPCT and WPPA (default options, column 2) gives the highest CS score for BAliBASE and the highest Q score for PREFAB. Based on this observation, the use of the two weighted approaches can be considered superior to that of the un-weighted ones, even though it gives a smaller SPS score for BAliBASE. In column 4, the use of PCT and WPPA gives the highest SPS and CS scores for BAliBASE, but produces a poorer Q score for PREFAB. Considering all these observations, our selection of default options is a trade-off between different benchmarks. To obtain high accuracy for PREFAB without significantly reducing the accuracy for BAliBASE is the main reason for using the two weighted approaches as default options. When comparing columns 3 and 4, for BAliBASE, WPPA seems to contribute more to the gain of accuracy improvement, but for PREFAB, WPCT seems to be better. Hence, we can say that the contribution of either WPCT or WPPA is dependent on the specific datasets. From the above observations and discussions, we can conclude that our combination of the two probabilistic models is a powerful approach to alignment accuracy improvement, and the two weighted approaches, as auxiliary features, contribute to the performance maximization as well.

4 DISCUSSION

We have presented MSAProbs, a new and practical algorithm for multiple protein sequence alignment designed based on pair-HMM and partition function posterior probabilities. On the four popular benchmark data sets including BAliBASE, PREFAB, SABmark and OXBENCH, MSAProbs demonstrates dramatic alignment accuracy improvements over several top performing aligners: ClustalW, MAFFT, MUSCLE, ProbCons and Probalign. Three strategies contribute most to accuracy improvement: the posterior probability matrix computation using pair-HMM and partition function posterior probabilities, the weighted probabilistic consistency transformation and the weighted profile–profile alignment. To reduce execution time, MSAProbs is further optimized for multi-core CPUs, as multi-core CPUs have become commonplace, by employing a multi-threaded design using OpenMP.

In addition to multiple protein sequence alignment, other issues in bioinformatics and computational biology, such as motif finding, RNA or protein structural prediction, might be able to benefit from our approaches.

ACKNOWLEDGEMENTS

We thank Robert C. Edgar for helpful discussion about issues on multiple protein sequence alignment benchmarking, and for providing the benchmark datasets and QSCORE scoring software.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Bahr,A. (2001) BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic Acids Res.*, **29**, 323–326.

Barton,G.J. and Sternberg,M.J. (1987) A strategy for the rapid multiple alignment of protein sequences. confidence levels from tertiary structure comparisons. *J. Mol. Biol.*, **198**, 327–337.

Berger,M.P. and Munson,P.J. (1991) A novel randomized iterative strategy for aligning multiple protein sequences. *Bioinformatics*, **7**, 479–484.

Boutonnet,N.S. *et al*. (1995) Optimal protein structure alignments by multiple linkage clustering: application to distantly related proteins. *Protein Eng.*, **8**, 647–662.

Brenner,S.E. *et al*. (2000) The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Res.*, **28**, 254–256.

Corpet,F. (1988) Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res.*, **16**, 10881–10890.

Do,C.B. *et al*. (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **15**, 330–340.

Durbin,R. *et al*. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK.

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

Edgar,R.C. (2004b) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 113.

Edgar,R.C. (2010) Quality measures for protein alignment benchmarks. *Nucleic Acids Res.* [Epub ahead of print; doi:10.1093/nar/gkp1196].

Feng,D.F. and Doolittle,R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, **25**, 351–361.

Gerstein,M. *et al*. (1994) Volume changes in protein evolution. *J. Mol. Biol.*, **236**, 1067–1078.

Gonnet,G.H. *et al*. (1992) Exhaustive matching of the entire protein sequence database. *Science*, **256**, 1443–1445.

Gotoh,O. (1995) A weighting system and algorithm for aligning many phylogenetically related sequences. *Comput. Appl. Biosci.*, **11**, 543–551.

Henikoff,S. and Henikoff,J.G. (1994) Position-based sequence weights. *J. Mol. Biol.*, **243**, 574–578.

Holm,L. and Sander,C. (1998) Touring protein fold space with Dali/FSSP. *Nucleic Acids Res.*, **26**, 316–319.

Katoh,K. *et al*. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, **30**, 3059–3066.

Katoh,K. *et al*. (2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res.*, **33**, 511–518.

Krogh,A. *et al*. (1994) Hidden markov models in computational biology: applications to protein modeling. *J. Mol.Biol.*, **235**, 1503–1531.

Larkin,M.A. *et al*. (2007) Clustal W and Clustal X version 2.0. *Bioinformatics*, **23**, 2948–2948.

Miyazawa,S. (1995) A reliable sequence alignment method based on probabilities of residue correspondences. *Protein Eng.*, **8**, 999–1009.

Notredame,C. *et al*. (1998) COFFEE: an objective function for multiple sequence alignments. *Bioinformatics*, **14**, 407–422.

Notredame,C. *et al*. (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.

OpenMP (2010) OpenMP tutorial. https://computing.llnl.gov/tutorials/openMP.

Rabiner,L.R. (1989) A tutorial on hidden markov models and selected applications in speech recognition. *In Proceedings of the IEEE*, **77**, 257–286.

Raghava,G.P. *et al*. (2003) OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics*, **4**, 47.

Roshan,U. and Livesay,D.R. (2006) Probalign: multiple sequence alignment using partition function posterior probabilities. *Bioinformatics*, **22**, 2715–2721.

Russell,R.B. and Barton,G.J. (1992) Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*, **14**, 309–323.

Saitou,N. and Nei,M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**, 406–425.

Shindyalov,I.N. and Bourne,P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, **11**, 739–747.

Siddiqui,A.S. *et al*. (2001) 3Dee: a database of protein structural domains. *Bioinformatics*, **17**, 200–201.

Sneath,P.H.A. and Sokal,R.P. (1973) *Numerical taxonomy*. Freeman, San Francisco, USA.

Studier,J.A. and Keppler,K.J. (1988) A note on the neighbor-joining method of Saitou and Nei. *Mol. Biol. Evol.*, **5**, 729–731.

Subbiah,S. and Harrison,S.C. (1989) A method for multiple sequence alignment with gaps. *J. Mol. Biol.*, **209**, 539–548.

Thompson,J.D. *et al*. (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. *et al*. (1999) BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, **15**, 87–88.

Thompson, J.D. *et al*. (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*, **61**, 127–136.

Van Walle,I. *et al*. (2004) Align-m–a new algorithm for multiple alignment of highly divergent sequences. *Bioinformatics*, **20**, 1428–1435.

Wilcoxon,F. (1947). Probability tables for individual comparisons by ranking methods. *Biometrics*, **3**, 119–122.