Sequence alignment by passing messages

Byung-Jun Yoon* [1]

[1]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA

Email: Byung-Jun Yoon*- bjyoon@ece.tamu.edu;

*Corresponding author

Abstract

Background: Sequence alignment has become an indispensable tool in modern molecular biology research, and probabilistic sequence alignment models have been shown to provide an effective framework for building accurate sequence alignment tools. One such example is the pair hidden Markov model (pair-HMM), which has been especially popular in comparative sequence analysis for several reasons, including their effectiveness in modeling and detecting sequence homology, model simplicity, and the existence of efficient algorithms for applying the model to sequence alignment problems. However, despite these advantages, pair-HMMs also have a number of practical limitations that may degrade their alignment performance or render them unsuitable for certain alignment tasks.

Results: In this work, we propose a novel scheme for comparing and aligning biological sequences that can effectively address the shortcomings of the traditional pair-HMMs. The proposed scheme is based on a simple message-passing approach, where messages are exchanged between neighboring symbol pairs that may be potentially aligned in the optimal sequence alignment. The message-passing process yields probabilistic symbol alignment confidence scores, which may be used for predicting the optimal alignment that maximizes the expected number of correctly aligned symbol pairs.

Conclusions: Extensive performance evaluation on protein alignment benchmark datasets shows that the proposed message-passing scheme clearly outperforms the traditional pair-HMM-based approach, in terms of both alignment accuracy and computational efficiency. Furthermore, the proposed scheme is numerically robust and amenable to massive parallelization.

Background

Sequence alignment has become an indispensable tool in modern molecular biology research, as it provides an effective and intuitive way of comparing and analyzing biological sequences. Given a set of biological sequences, the primary objective of sequence alignment is to predict the best overall mapping between the sequences, which accurately aligns the homologous regions that are embedded in them. This provides an effective means for detecting conserved sequence regions with potentially important functional roles. The concept of sequence alignment has had diverse applications in biomedical research [1–7], which include homology search, function and structure prediction of biomolecules, phylogenetic analysis, and detecting sequence motifs, among others.

Typically, sequence alignment is carried out by formulating and solving an optimization problem – either implicitly or explicitly – where the goal is to maximize an objective function that measures the overall quality of the sequence alignment. For example, one simple way of aligning a sequence pair would be to score each potential alignment by assigning a "substitution score" to every aligned symbol pair and penalty scores for gaps and then find the optimal alignment that maximizes the overall score through dynamic programming [1]. In the past, various *ad hoc* scoring schemes have been proposed to obtain intuitive and biologically meaningful sequence alignment results. As an alternative to heuristic scoring schemes, there have been also research efforts to develop probabilistic models for sequence alignment that can be used to evaluate and compare potential alignments and to estimate the symbol-to-symbol alignment probabilities.

Examples of such probabilistic schemes include the pair hidden Markov models (pair-HMMs) [1] and the partition function based scheme [8]. Given two biological sequences, these methods can be used to estimate the posterior symbol alignment probability for each symbol pair that may be aligned in the final sequence alignment. Based on the estimated probabilities, we can predict the optimal sequence alignment that contains the largest expected number of correctly aligned symbol pairs, rather than an alignment that maximizes an *ad hoc* score. This is typically referred to as the maximum expected accuracy (MEA) alignment [9–11], and as before, it can be also found through dynamic programming.

Among a number of probabilistic sequence alignment models, pair-HMMs have been especially popular, and they have been widely adopted by many multiple sequence alignment (MSA) algorithms, including ProbCons [9] and PicXAA [10]. Despite the simplicity of the model, pair-HMMs have been shown to be very effective in modeling sequence homology, as reflected in the well-rounded overall performance of various MSA algorithms that utilize the symbol alignment probabilities estimated by pair-HMMs. Furthermore, these probabilities can be estimated in a relatively efficient manner, making the pair-HMMs an attractive

choice for various sequence alignment problems. However, pair-HMMs also have a number of shortcomings, which may negatively affect their alignment performance or make them impractical for certain alignment tasks.

In this paper, we propose a novel scheme for comparing and aligning biological sequences that can effectively address the limitations of pair-HMMs. The proposed scheme computes probabilistic symbol alignment confidence scores based on a simple and computationally efficient message-passing approach. As we will demonstrate in this paper, this message-passing scheme has a number of important advantages over the traditional pair-HMMs and it clearly outperforms pair-HMMs in terms of both speed and accuracy on protein alignment benchmark datasets.

Methods

A brief overview of pair hidden Markov models

The pair-HMM [1, 12] is a generative sequence model that can simultaneously generate a *pair* of aligned symbol sequences. This is different from the traditional HMMs, which generate only a single symbol sequence at a time [13]. Figure 1 shows two examples of pair-HMMs that are widely used in biological sequence analysis. As shown in Figure 1, a typical pair-HMM consists of three hidden states I_x, I_y, and M, which

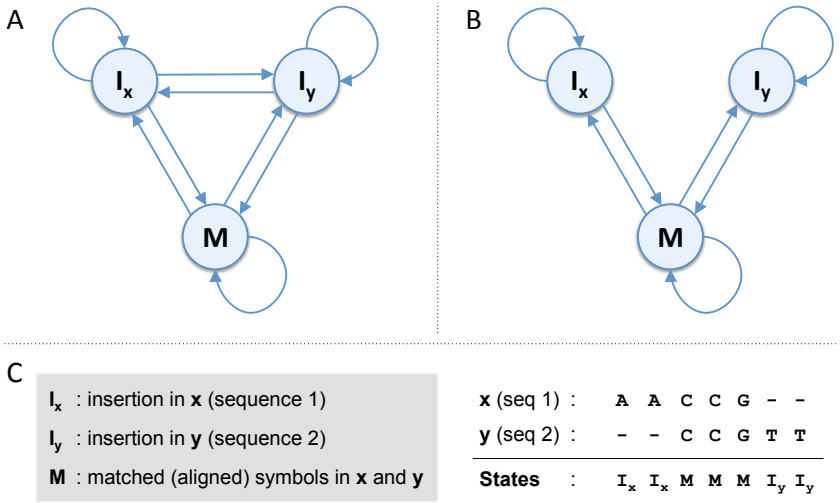

Figure 1: **Pair hidden Markov models.** (A) The state transition diagram of a widely used pair-HMM. (B) An alternative pair-HMM implementation that does not allow transitions between the two insertion states I_x and I_y. (C) An example of a sequence pair $(\mathbf{x}, \mathbf{y})$ that is generated by a pair-HMM.

are used to model insertions in sequence $\mathbf{x}$, insertions in sequence $\mathbf{y}$, and matched (i.e., aligned) symbols in both sequences, respectively. The pair-HMM generates an aligned sequence pair $(\mathbf{x}, \mathbf{y})$ by making transitions between the hidden states according to the specified state transition probabilities. At state I_x, the model emits a symbol only to sequence $\mathbf{x}$, while at I_y, a symbol is emitted only to sequence $\mathbf{y}$. On the other hand, at state M, the model emits a pair of aligned symbols, where one symbol is added to $\mathbf{x}$ and the other symbol is added to $\mathbf{y}$. Figure 1(C) gives an example of a sequence pair $(\mathbf{x}, \mathbf{y})$ that is generated by a pair-HMM. In this example, the underlying hidden state sequence that gives rise to the two sequences $\mathbf{x} = \mathsf{AACCG}$ and $\mathbf{y} = \mathsf{CCGTT}$ is $\mathsf{I_xI_xMMMI_yI_y}$. This indicates that the first two symbols (i.e., AA) in $\mathbf{x}$ and the last two symbols in $\mathbf{y}$ (i.e., TT) are "insertions," which do not have any matching counterpart in the other sequence, while the last three symbols in $\mathbf{x}$ and the first three symbols in $\mathbf{y}$ (i.e., CCG in both sequences) are jointly generated by the pair-HMM, hence closely match each other. As we can see from this example, we can unambiguously identify the alignment of a given sequence pair $(\mathbf{x}, \mathbf{y})$, once the underlying hidden state sequence yielding the sequence pair is known. Of course, the hidden state sequence is generally not known, but there exist efficient algorithms that can be used for its prediction. For example, we can use the Viterbi algorithm [14] to predict the optimal hidden state sequence that maximizes the observation probability of the sequence pair $(\mathbf{x}, \mathbf{y})$. Alternatively, we can also predict the state sequence that maximizes the expected number of correct states, by first estimating the alignment probabilities between the symbols in $\mathbf{x}$ and $\mathbf{y}$ through the forward and backward procedures [13] and then applying the Needleman-Wunsch algorithm [15]. This will lead to the MEA alignment between the two sequences $\mathbf{x}$ and $\mathbf{y}$.

Limitations of pair-HMMs

Although the hidden state sequence of a pair-HMM unambiguously points to a specific sequence alignment, this is not necessarily true the other way around. In fact, several different state sequences can lead to the same sequence alignment, hence we may not always be able to unambiguously determine the underlying state sequence for a given pairwise sequence alignment. For example, let us consider two sequences $\mathbf{x} = \mathsf{AAACGG}$ and $\mathbf{y} = \mathsf{AAATTA}$. Suppose the "true" alignment aligns only the first three symbols (i.e., AAA) of $\mathbf{x}$ and $\mathbf{y}$, hence the last three symbols in the respective sequences are regarded as insertions that do not have any matching counterpart in the other sequence. This is illustrated below, where the solid lines correspond to the aligned symbols:

$$\begin{array}{lcccccc} \mathbf{x}: & \mathsf{A} & \mathsf{A} & \mathsf{A} & \mathsf{C} & \mathsf{G} & \mathsf{G} \\ & | & | & | & & & \\ \mathbf{y}: & \mathsf{A} & \mathsf{A} & \mathsf{A} & \mathsf{T} & \mathsf{T} & \mathsf{A} \end{array} \tag{1}$$

For the pair-HMM shown in Figure 1(A), any hidden state sequence $\mathbf{s} = s_1 s_2 \cdots s_9$ such that $s_1 = s_2 = s_3 = \mathsf{M}$ and $s_4 s_5 \cdots s_9$ is a permutation of $\mathsf{I_x}\,\mathsf{I_x}\,\mathsf{I_x}\,\mathsf{I_y}\,\mathsf{I_y}\,\mathsf{I_y}$ would lead to the sequence alignment shown in (1). When using this pair-HMM for predicting the optimal alignment of a sequence pair with the largest probability, this ambiguity may lead to performance degradation as these potential state sequences compete against each other. For this reason, it is generally more desirable to estimate the symbol alignment probabilities via the pair-HMM by considering all potential alignments and state sequences and use the estimated probabilities to find the MEA alignment that is expected to have the maximum number of correctly aligned symbols [9–11]. However, the aforementioned ambiguity also negatively affects the quality of the estimated symbol alignment probabilities, which is especially noticeable for sequence pairs with low percentage identity. In some cases, the alternative pair-HMM shown in Figure 1(B) is used to avoid such ambiguity. This alternative pair-HMM blocks transitions between the insertion states $\mathsf{I_x}$ and $\mathsf{I_y}$, thereby prohibiting the model from inserting unaligned symbols to both sequences. For example, the alignment shown in (1) would not be allowed based on this alternative pair-HMM. However, due to this restriction, the pair-HMM in Figure 1(B) has a relatively stronger tendency to align unrelated sequence regions by treating them as mutations. This may again negatively affect the quality of the symbol alignment probabilities estimated based on the pair-HMM.

Another potential drawback of pair-HMMs is that the associated algorithms (i.e., the Viterbi, forward, and backward algorithms) can become numerically unstable for long sequences. Application of pair-HMMs to biological sequence analysis involves computing extremely small probabilities, which decrease exponentially with the sequence length. For example, based on the pair-HMM that was used in [9], the observation probability (i.e., the probability that the HMM may generate a given sequence pair) of a protein pair is typically in the order of 10^{-230} for proteins of length 80, 10^{-280} for proteins of length 100, and 10^{-320} for proteins of length 120. As a result, pair-HMM algorithms are prone to underflow errors, unless they are carefully implemented to keep them numerically robust. So far, a number of schemes have been proposed to address this issue, such as using log transformations of the probabilities or normalizing the probabilities to keep them within a reasonable numerical range, and have been shown to work well for relatively long sequences [1]. However, log transformations can make the forward and backward algorithms considerably slower, and the normalization approach can still lead to underflow errors as the sequences get longer.

One further disadvantage of pair-HMMs is that the algorithms that are used with the model cannot be easily parallelized. Although the Viterbi, forward, and backward algorithms for pair-HMMs are relatively efficient, they are still computationally expensive to be used with very long sequences. Moreover, as the algorithms are not amenable to massive parallelization, this makes the pair-HMMs not suitable for large-scale

sequence analysis tasks, such as the whole genome alignment, despite their superior performance compared to other heuristic methods.

A message-passing scheme for estimating symbol alignment confidence scores

Here, we propose a novel method for aligning biological sequences that can effectively address the aforementioned shortcomings of pair-HMMs. The proposed method is based on a message-passing scheme, where messages are iteratively exchanged between neighboring symbol pairs to estimate the level of confidence for potential pairwise symbol alignments. The main underlying motivation is to develop an "analytical" method that can *directly* estimate the symbol alignment probabilities, without specifically modeling symbol insertions and deletions. This stands in contrast to the pair-HMM approach, which is essentially based on a "generative" sequence model that tries to explicitly model symbol insertions/deletions, in addition to symbol alignments. As discussed before, modeling symbol insertions in pair-HMMs can lead to subtle issues with potentially negative effects, and considering that our ultimate goal lies in finding an accurate sequence alignment through effective estimation of the symbol alignment probabilities, a method that can directly estimate these probabilities without explicitly modeling insertions/deletions would be desirable.

Suppose $\mathbf{x} = x_1 x_2 \cdots x_L$ and $\mathbf{y} = y_1 y_2 \cdots y_M$ are the two sequences to be aligned. We define $c_{\mathbf{xy}}(i, j)$

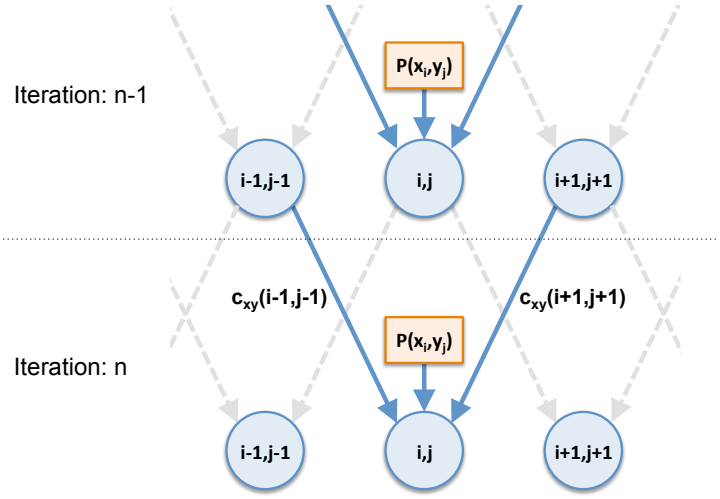

Figure 2: **Illustration of the proposed message-passing scheme.** At iteration n, the alignment confidence score $c_{\mathbf{xy}}(i, j)$ of the symbol pair (x_i, y_j) is updated based on the messages received from its neighbors (x_{i-1}, y_{j-1}) and (x_{i+1}, y_{j+1}) and the joint occurrence probability $P(x_i, y_j)$ of the symbols x_i and y_j. Solid lines indicate the messages that are used to update $c_{\mathbf{xy}}(i, j)$, while the dashed lines correspond to messages that are used to update the alignment confidence scores of other symbol pairs.

as the *symbol alignment confidence score* between x_i (the i-th symbol in $\mathbf{x}$) and y_j (the j-th symbol in $\mathbf{y}$). The score $c_{\mathbf{xy}}(i,j)$ provides a quantitative measure of confidence as to whether x_i and y_j should be aligned to each other or not, and we assume $c_{\mathbf{xy}}(i,j) \propto P(x_i \sim y_j|\mathbf{x},\mathbf{y})$, where $P(x_i \sim y_j|\mathbf{x},\mathbf{y})$ is the posterior symbol alignment probability between x_i and y_j given the sequences $\mathbf{x}$ and $\mathbf{y}$. We estimate the alignment confidence score by iteratively passing messages between neighboring symbol pairs, where each symbol pair (x_i, y_j) corresponds to a potential symbol alignment in the true (unknown) sequence alignment between $\mathbf{x}$ and $\mathbf{y}$. For example, during the estimation process, the symbol pair (x_i, y_j) will exchange messages with its two neighbors (x_{i-1}, y_{j-1}) and (x_{i+1}, y_{j+1}), and similarly, the pair (x_{i+1}, y_{j+1}) will exchange messages with (x_i, y_j) and (x_{i+2}, y_{j+2}). The message-passing process is illustrated in Figure 2, where the solid lines indicate the messages that are used to update the alignment confidence score $c_{\mathbf{xy}}(i,j)$ of the symbol pair (x_i, y_j). The dashed lines correspond to messages that are used to update the confidence scores of other symbol pairs.

The pseudocode of the proposed message-passing algorithm is as follows:

`STEP-1` `Initialize` $c_{\mathbf{xy}}(i,j)$.

`STEP-2` `Update the alignment confidence score:`

$$c_{\mathbf{xy}}(i,j) \longleftarrow \lambda \left\{ \frac{c_{\mathbf{xy}}(i-1,j-1)+c_{\mathbf{xy}}(i+1,j+1)}{2} \right\} + (1-\lambda) P(x_i, y_j).$$

`STEP-3` `Normalize` $c_{\mathbf{xy}}(i,j)$.

`STEP-4` `If` $c_{\mathbf{xy}}(i,j)$ `has converged, then terminate the algorithm.`
`Otherwise, go to STEP-2.`

In `STEP-1`, we first initialize the alignment confidence score $c_{\mathbf{xy}}(i,j)$, where we can simply use random initialization. If a preliminary sequence alignment of $\mathbf{x}$ and $\mathbf{y}$ is available (e.g., obtained from a simple heuristic method), we can also initialize the score based on this alignment such that $c_{\mathbf{xy}}(i,j) = 1$ if x_i and y_j are aligned, and $c_{\mathbf{xy}}(i,j) = 0$ otherwise. Next, in `STEP-2`, the alignment confidence score $c_{\mathbf{xy}}(i,j)$ of the symbol pair (x_i, y_j) is updated based on the scores of its two neighbors (x_{i-1}, y_{j-1}) and (x_{i+1}, y_{j+1}). Note that the score is set to $c_{\mathbf{xy}}(i,j) = 0$ if $i \notin \{1, \cdots, L\}$ or $j \notin \{1, \cdots, M\}$. $P(x_i, y_j)$ is the joint occurrence probability of the symbol pair (x_i, y_j), which is essentially equivalent to the joint emission probability of an aligned symbol pair (x_i, y_j) at the match state M of a pair-HMM. It should be noted that this probability $P(x_i, y_j)$ is not location-dependent and is simply determined by the symbols x_i and y_j. The weight parameter $\lambda \in [0, 1]$ is used to balance the contribution from the neighbors and that from the joint probability of (x_i, y_j) in estimating the alignment confidence score. A large λ gives more weight to the "messages" received from

the neighbors in estimating the scores, which tends to penalize gaps more heavily, and it generally leads to longer aligned regions with fewer gaps. On the contrary, a small λ gives more weight to the joint symbol occurrence probability $P(x_i, y_j)$ while giving less weight to the messages received from the neighbors, which tends to be more lenient to gaps. Once the symbol alignment confidence score $c_{\mathbf{xy}}(i, j)$ is updated for all $i = 1, \cdots, L$ and $j = 1, \cdots, M$, we normalize the scores to keep them within a proper numerical range, as shown in `STEP-3`. For example, a simple way would be to divide the score matrix $\mathbf{C} = \left[c_{\mathbf{xy}}(i, j)\right]$ by its matrix norm to normalize the confidence scores. After normalization, the updated scores are compared to the scores in the last iteration, and the algorithm terminates if the specified convergence criterion has been met. Otherwise, the algorithm goes back to `STEP-2` and repeats the message-passing process.

Results and Discussion

Dataset and experimental set-up

In order to evaluate the performance of the proposed message-passing scheme, we carried out pairwise sequence alignment experiments based on the BAliBASE 3.0 protein alignment benchmark [16]. BAliBASE is arguably the most widely used benchmark for multiple sequence alignment, and it has been utilized by most multiple sequence alignment algorithms for assessing their performance. The benchmark consists of five reference sets, where Reference 1 consists of two subsets: V1 and V2. Each reference set consists of multiple sequence alignments that satisfy specific criteria, such that different reference sets can be used to test the performance of multiple sequence alignment algorithms under different conditions. For example, each alignment in Reference 2 consists of sequences that share reasonably high identity ($> 40\%$) and "orphan sequences" that share little identity ($< 20\%$) to other sequences in the alignment. Reference sets 4 and 5 are constructed such that every sequence has at least one other sequence in the same alignment whose identity exceeds 20%. Sequences in Reference 4 and Reference 5 may contain large N/C-terminal extensions or internal insertions, respectively. Further details of the BAliBASE 3.0 benchmark can be found in [16].

For every sequence family in BAliBASE 3.0, we performed pairwise sequence alignment for all possible sequence pairs in the given family. The pairwise alignment was performed in the following manner. First, we estimated the probabilistic symbol alignment confidence score using the proposed message-passing scheme. In our experiments, we used three different values of λ (=0.25, 0.5, and 0.75) to investigate the effect of λ on the overall sequence alignment performance. For the joint symbol occurrence probability $P(x_i, y_j)$, we used the joint emission probability (at state M) of the pair-HMM that was used in [9]. At the end of each iteration, we normalized the alignment confidence score by dividing the confidence score matrix $\mathbf{C}$ by the matrix 2-

norm: $\mathbf{C} \longleftarrow \mathbf{C}/\|\mathbf{C}\|_2$. We terminated the message-passing process if $\sum_i \sum_j \left| c_{\mathbf{xy}}(i,j) - \widetilde{c}_{\mathbf{xy}}(i,j) \right| < 0.01$, where $c_{\mathbf{xy}}(i,j)$ is the current score and $\widetilde{c}_{\mathbf{xy}}(i,j)$ is the score obtained in the previous iteration. Once the scores converged, based on our assumption that $c_{\mathbf{xy}}(i,j) \propto P(x_i \sim y_j|\mathbf{x},\mathbf{y})$, we used the confidence score $c_{\mathbf{xy}}(i,j)$ to find the MEA alignment through dynamic programming. The predicted alignment was compared to the benchmark alignment in BAliBASE 3.0 to compute the sensitivity (SN) = $\frac{TP}{TP+FN}$ and the positive predictive value (PPV) = $\frac{TP}{TP+FP}$, where TP is the number of correctly aligned symbol pairs, FP is the number of incorrectly aligned pairs, and FN is the number of symbol pairs that are aligned in the benchmark alignment but not aligned in the predicted alignment. For comparison, we repeated similar experiments using the pair-HMM with the same set of parameters as the one used in [9].

Performance of the proposed message-passing scheme

Table 1 summarizes the pairwise sequence alignment performance of the proposed message-passing scheme and the traditional pair-HMM approach. Each row shows the evaluation results on each of the six reference sets (i.e., RV11, RV12, RV20, RV30, RV40, RV50) in BAliBASE 3.0. For each reference set, we estimated the average SN, PPV, and CPU time (for estimating the alignment scores/probabilities) of different alignment schemes based on all possible pairwise sequence alignments: 943 alignments for the reference set RV11, 2,335 alignments for RV12, 50,062 alignments for RV20, 76,370 alignments for RV30, 23,445 alignments for RV40, and 7,538 alignments for RV50. All experiments were performed using Matlab on a MacPro workstation with two 2.8 GHz Quad-Core Intel Xeon processors and 32GB memory.

From Table 1, we can clearly see that the proposed message-passing scheme significantly outperforms the

Table 1: **Pairwise sequence alignment performance evaluated on the BAliBASE 3.0 benchmark.**

Ref	**Pair-HMM**			**Message-Passing**								
				$\lambda = 0.25$			$\lambda = 0.50$			$\lambda = 0.75$		
	SN	PPV	CPU	SN	PPV	CPU	SN	PPV	CPU	SN	PPV	CPU
RV11	0.048	0.106	1.934	0.123	0.149	**0.769**	0.155	0.175	1.465	**0.198**	**0.209**	3.675
RV12	0.213	0.414	2.707	0.399	0.468	**1.146**	0.475	0.523	2.145	**0.569**	**0.595**	5.200
RV20	0.276	0.476	2.725	0.504	0.568	**1.186**	0.570	0.613	2.251	**0.643**	**0.665**	5.531
RV30	0.168	0.300	2.656	0.324	0.369	**1.143**	0.372	0.402	2.160	**0.426**	**0.441**	5.432
RV40	0.153	0.271	4.084	0.250	0.284	**1.760**	0.300	0.323	3.234	**0.361**	**0.373**	7.970
RV50	0.140	0.254	4.969	0.248	0.278	**2.102**	0.294	0.312	3.967	**0.348**	**0.353**	9.856

The average sensitivity (SN), positive predictive value (PPV), and CPU time (seconds) on different reference sets are shown for each sequence alignment scheme. All experiments were performed in Matlab on a MacPro workstation with 2×2.8 GHz Quad-Core Intel Xeon processors and 32GB memory.

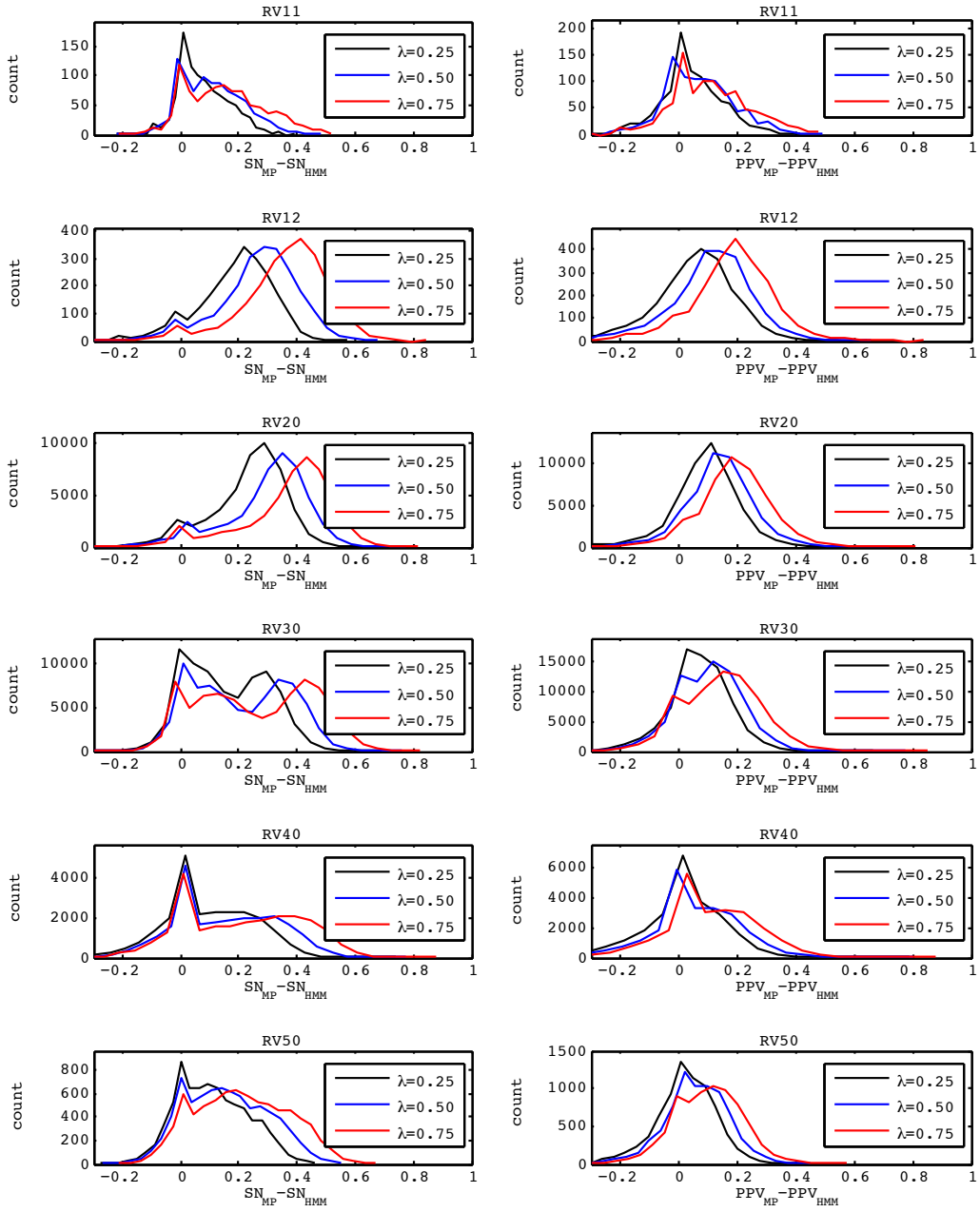

Figure 3: **Performance comparison between the proposed message-passing scheme and the traditional pair-HMM approach.** The plots in the left column show the distributions of the sensitivity difference $SN_{MP} - SN_{HMM}$ between the message-passing scheme and the pair-HMM-based approach. In the right column, the distributions of the difference between the positive predictive values $PPV_{MP} - PPV_{HMM}$ of the two schemes are shown. Each row shows the evaluation results obtained from each of the six reference sets in BAliBASE 3.0.

pair-HMM approach in terms of SN and PPV, for all three values of λ. For example, the message-passing scheme achieved up to 0.23 higher SN and 0.09 higher PPV for $\lambda = 0.25$, and up to 0.37 higher SN and 0.19 higher PPV for $\lambda = 0.75$. Our experiments showed that a larger λ tends to yield more accurate alignments, while a smaller λ tends to make the algorithm converge faster, hence computationally more efficient. For example, when the weight parameter was set to $\lambda = 0.25$, the message-passing scheme was around $2.3 \sim 2.5$ times faster than the pair-HMM, while still yielding much more accurate alignments.

The results in Table 1 demonstrate that, on average, the proposed message-passing scheme considerably improves the quality of sequence alignment over the traditional pair-HMM approach. In order to see whether the proposed scheme also leads to a consistent improvement for most sequence pairs, we calculated the difference between SN_{MP} (the sensitivity of the message-passing scheme) and SN_{HMM} (the sensitivity of the pair-HMM-based approach) for every pairwise sequence alignment that we have performed in our experiments. Similarly, we calculated the difference between PPV_{MP} (the PPV of the message-passing scheme) and PPV_{HMM} (the sensitivity of the pair-HMM approach) for all sequence pairs in BAliBASE 3.0. Figure 3 shows the distributions of $SN_{MP} - SN_{HMM}$ and $PPV_{MP} - PPV_{HMM}$ for all sequence pairs. To avoid any bias from unsuccessful alignments, sequence pairs for which neither method yielded an alignment with at least one correct symbol alignment were excluded. The plots in the left column of Figure 3 show the distributions of $SN_{MP} - SN_{HMM}$, and those in the right column show the distributions of $PPV_{MP} - PPV_{HMM}$. The results obtained from the same reference set are shown in the same row, where the first row shows the results on RV11 and the last row shows the results on RV50. As we can see in Figure 3, every single distribution shown in the figure has a much larger probability mass in the right-half plane, which clearly demonstrates that the proposed message-passing scheme consistently outperforms the pair-HMM-based approach for most (though not all) sequence pairs. In many cases, the improvements in SN and PPV were quite significant ($0.4 \sim 0.8$), which shows that the proposed scheme can often find an accurate sequence alignment even when the pair-HMM has difficulty aligning the sequences.

Conclusions

In this paper, we proposed a novel method for sequence alignment based on an efficient message-passing approach. Given two biological sequences, the proposed method estimates the symbol alignment confidence scores for all possible symbol pairs. These scores are iteratively computed by exchanging messages between neighboring symbol pairs, where empirical evidence shows that these scores quickly converge within several iterations. The proposed message-passing scheme effectively addresses a number of limitations of the tradi-

tional pair-HMM-based approach, and extensive performance assessment based on BAliBASE 3.0 shows that the proposed scheme consistently outperforms the pair-HMM approach, both in terms of alignment accuracy and computational efficiency. Considering that pair-HMMs have been widely adopted by many modern multiple sequence alignment algorithms [9–11], the proposed scheme has potentials to further improve the current state-of-the-art. Furthermore, the proposed scheme is numerically stable even for extremely long sequences. Unlike the pair-HMM approach, there is no global measure or quantity (such as the observation probability $P(\mathbf{x}, \mathbf{y})$ of the entire sequence pair) to be estimated, and the exchanged messages (i.e., symbol alignment confidence scores) are normalized after each iteration, which ensures that they lie within a reasonable numerical range. Finally, the simple iterative estimation process – in which the neighboring symbol pairs only exchange "local" messages – makes the proposed message passing scheme amenable to massive parallelization through the utilization of modern GPU (graphics processing unit) architecture. These characteristics open up the possibility of applying the proposed message-passing scheme to accurate probabilistic alignment of genome-scale sequences, which has not been possible using traditional pair-HMMs.

Finally, it is worth noting that the formula that is used to update $c_{\mathbf{xy}}(i, j)$ in the proposed message-passing algorithm bears conceptual similarity to the eigenvalue equation used by the network alignment algorithm called IsoRank [17] for estimating the functional similarity between proteins across different protein-protein interaction (PPI) networks. As demonstrated in [18, 19], techniques that were originally developed for sequence alignment may also have potentials to improve network alignment methods. Conversely, techniques used in network alignment may also lead to better sequence alignment methods. For example, the scoring scheme used by IsoRank can be viewed as a random walk [20], and it was shown that the use of a different random walk scheme can lead to more accurate network alignment results [19]. Similarly, it may be possible to modify the update formula for $c_{\mathbf{xy}}(i, j)$ to further improve the performance of the proposed message-passing scheme, and we are currently in the process of investigating several different implementations.

Author's contributions

BJY conceived the idea, performed the simulations, analyzed the results, and wrote the paper.

Competing interests

The author declares that he has no competing interests.

Acknowledgments

This work was supported by the National Science Foundation through NSF Award CCF-1149544.

Declarations

Publication of this article was funded by the National Science Foundation through NSF Award CCF-1149544.

References

1. Durbin R, Eddy SR, Krogh A, Mitchison G: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press 1998.
2. Phillips A, Janies D, Wheeler W: **Multiple sequence alignment in phylogenetic analysis**. *Mol. Phylogenet. Evol.* 2000, **16**:317–330.
3. Cuff JA, Barton GJ: **Application of multiple sequence alignment profiles to improve protein secondary structure prediction**. *Proteins* 2000, **40**:502–511.
4. Notredame C: **Recent progress in multiple sequence alignment: a survey**. *Pharmacogenomics* 2002, **3**:131–144.
5. Edgar RC, Batzoglou S: **Multiple sequence alignment**. *Curr. Opin. Struct. Biol.* 2006, **16**:368–373.
6. Pei J: **Multiple protein sequence alignment**. *Curr. Opin. Struct. Biol.* 2008, **18**:382–386.
7. Kumar S, Filipski A: **Multiple sequence alignment: in pursuit of homologous DNA positions**. *Genome Res.* 2007, **17**:127–135.
8. Miyazawa S: **A reliable sequence alignment method based on probabilities of residue correspondences**. *Protein Eng.* 1995, **8**(10):999–1009.
9. Do CB, Mahabhashyam MS, Brudno M, Batzoglou S: **ProbCons: Probabilistic consistency-based multiple sequence alignment**. *Genome Res.* 2005, **15**(2):330–340.
10. Sahraeian SM, Yoon BJ: **PicXAA: greedy probabilistic construction of maximum expected accuracy alignment of multiple sequences**. *Nucleic Acids Res.* 2010, **38**(15):4917–4928.
11. Hamada M, Asai K: **A classification of bioinformatics algorithms from the viewpoint of maximizing expected accuracy (MEA)**. *J. Comput. Biol.* 2012, **19**(5):532–549.
12. Yoon BJ: **Hidden Markov models and their applications in biological sequence analysis**. *Curr. Genomics* 2009, **10**(6):402–415.
13. Rabiner LR: **A tutorial on hidden Markov models and selected applications in speech recognition**. *Proceedings of the IEEE* 1989, **77**(2):257–286.
14. Viterbi A: **Error bounds for convolutional codes and an asymptotically optimum decoding algorithm**. *Information Theory, IEEE Transactions on* 1967, **13**(2):260–269.
15. Needleman SB, Wunsch CD: **A general method applicable to the search for similarities in the amino acid sequence of two proteins**. *J. Mol. Biol.* 1970, **48**(3):443–453.
16. Thompson JD, Koehl P, Ripp R, Poch O: **BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark**. *Proteins* 2005, **61**:127–136.
17. Singh R, Xu J, Berger B: **Global alignment of multiple protein interaction networks with application to functional orthology detection**. *Proc. Natl. Acad. Sci. U.S.A.* 2008, **105**(35):12763–12768.
18. Flannick J, Novak A, Srinivasan BS, McAdams HH, Batzoglou S: **Graemlin: general and robust alignment of multiple large interaction networks**. *Genome Res.* 2006, **16**(9):1169–1181.
19. Sahraeian SM, Yoon BJ: **SMETANA: Accurate and Scalable Algorithm for Probabilistic Alignment of Large-Scale Biological Networks**. *PLoS ONE* 2013, **8**(7):e67995.
20. Yoon BJ, Qian X, Sahraeian SME: **Comparative analysis of biological networks: Hidden Markov model and Markov chain-based approach**. *IEEE Signal Processing Magazine* 2012, **29**:22–34.