Journal of Computational Intelligence in Bioinformatics
ISSN: 0973-385X Volume 3 Number 2 (2010) pp. 147–161

http://www.ripublication.com/jcib.htm

Global Suboptimal Pairwise Sequence Alignment in Linear Space Using Pair Hidden Markov Models

Fakhry M. Khellah

Computer Science Department, Prince Sultan University, Saudi Arabia
E-mail: fkhellah@cis.psu.edu.sa

Abstract

Global pairwise sequence alignment is an important tool in many bioinformatics applications. There are two main approaches for pairwise sequence alignment: the standard dynamic programming approach and the statistical alignment approach based on pair Hidden Markov models (HMM). Standard algorithms implementing both approaches suffer from the high storage demands needed to save the backtracking pointers needed to obtain the actual sequence alignment. There are several alternative algorithms that deal with the storage demands such as the alignment in linear space or using divide-and-conquer methods. However, these methods either do not provide the actual alignment or increase the implementation complicity. In this paper, we present a new technique for global sequence alignment based on combining the linear space implementation and Pair Hidden Markov models. We demonstrate how the proposed method can be applied to global pairwise alignment of DNA sequences, in addition to obtaining an approximate pair HMM for the two sequences. Since the traceback step does not exist in proposed approach, it may find application in the alignment of large biological sequences.

Keywords: Global Sequence Alignment , Hidden Markov Models, Linear Space.

Introduction

Alignment is the most basic component of biological sequence manipulation that is mainly used to discover and measure the similarity between biological sequences. It has several applications in sequence assembly, sequence annotation, structural and functional predictions for genes and proteins. Researchers have formulated a large number of various alignment problems that address different scientific goals including

local alignment, global alignment, multiple alignment, alignment of proteins, or nucleotides and many others [9], [12], [23].

In global pairwise optimal alignment, two sequences $X = x_1, \cdots, x_m$, and $Y = y_1, \cdots, y_n$, $(m \le n)$, are aligned entirely, i.e, from end-to-end. Generally, the global alignment problem is solved efficiently using the Needleman-Wunsch algorithm [21] and the more efficient version introduced by Gotoh [9]. In that work, the algorithm builds up an optimal alignment using the previous solutions for optimal alignments of smaller subsequences. The dynamic programming solution is based on using a matrix F of size $(m+1)\times(n+1)$, where in the horizontal and vertical directions lie the sequences X and Y, respectively. Each cell $F(i, j)$ in the dynamic programming matrix F represents the highest scoring alignment between the initial segments of both sequences, $(X[1\cdots i], Y[1\cdots j])$.

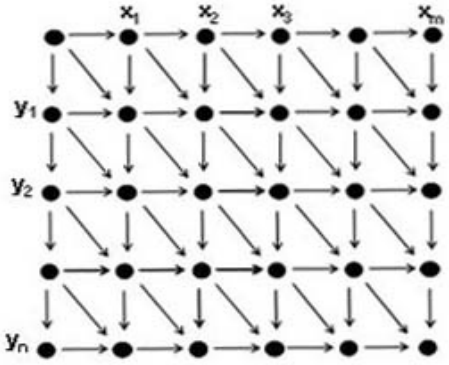

Figure 1: The two-dimensional dynamic programming alignment matrix used in the standard alignment algorithm. The value of each cell $F(i, j)$ is derived from one of its three neighbors: diagonal $F(i-1, j-1)$, left $F(i, j-1)$, and top $F(i-1, j)$.

The values of F are filled recursively starting from top left to bottom right. Each cell $F(i, j)$ is computed based on the values of its three immediate neighbors $F(i-1, j-1)$, $F(i, j-1)$, and $F(i-1, j)$ as shown in Figure (1).

There are three possible ways that the best score of an alignment up to residues x_i, y_i could be obtained. First, the bases x_i and y_i are aligned. In this case, the score is either increased if x_i, y_i are the same and decreased if they differ. Second, x_i is aligned with a gap (insertion). Third, y_i is aligned with a gap (deletion). In case of insertion and deletion the score is decreased by the imposed gap penalty. Therefore, the corresponding recurrence for the score $F(i, j)$ is given by:

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_i) \\ F(i-1, j) + \gamma \\ F(i, j-1) + \gamma \end{cases} \quad (1)$$

Where $s(x_i, y_i)$ is the score for each aligned residue pair and score γ is the imposed gap penalty. To reconstruct the best alignment, a pointer to the cell from which $F(i,j)$ was derived is stored. After the matrix F is computed, the score of the best global alignment of sequences X ,Y is given in the final cell of the matrix $F(m,n)$. In order to find the alignment itself, the dynamic programming matrix F must be traversed starting from the cell $F(m,n)$ to the origin of the matrix $F(0,0)$ in order to find the path(s) that lead to the maximum score according to choices made by (1). The procedure is called traceback and requires that the whole dynamic programming matrix be in memory [9], [12], [23]. The above algorithm requires $O(mn)$ in both *time* and *space*. Given the quadratic amount of memory demand of the algorithm, it becomes clear that the alignment of large sequences can be prohibitive.

Several techniques exist that can solve the dynamic programming alignment in linear space [5], [12]. The best alignment score can be computed in a linear storage requirement by saving just two columns of the dynamic programming matrix (i.e., the current column and the previous one) as shown in Figure (2). However, obtaining the actual optimal alignment of the two sequences is still a challenge because it requires that dynamic programming matrix be stored in memory in order not to lose the traceback pointers.

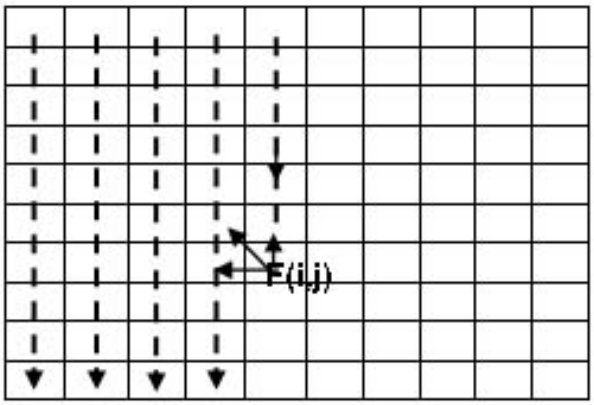

Figure 2: Illustration of the needed dynamic programming matrix columns for alignment in linear space. Observe that computing the alignment scores in each column requires only the scores in the preceding column.

In order to find the explicit global pairwise alignment in addition to the optimal score in linear space, a recursive divide and conquer procedure, introduced by Hirschberg and applied to biological sequence alignment problem by Myers and Miller [5], [7], [20] , is used. In the divide and conquer algorithm, the alignment problem is divided into two smaller subproblems and the whole path is found recursively. Although, this approach performs the alignment in linear space it doubles the computational time and increases the implementation complexity. Other heuristic approaches such as BLAST and FASTA and their variations [9], [12], [19] have been

used to overcome the time and space requirements of the standard dynamic programming algorithm. However, they are mainly used to perform local pairwise alignment. In addition, heuristic approaches do not guarantee to produce an optimal solution. The *key* to our proposed approach is the following: rather than storing the whole dynamic programming matrix needed by the traceback step in order to find the explicit alignment, a suboptimal alignment is then produced using a pair hidden Markov model (pair HMM). The pair HMM parameters are estimated while sweeping the dynamic programming matrix to compute the optimal global alignment score using the linear space approach.

The paper addresses the background on pair HMM in Section (2), followed by our proposed approach for global pairwise sequence alignment in Section (3). Finally, Section (4) presents examples on performing global pairwise alignment using the learned pair HMM model parameters.

Review of Sequence Alignment Using Pair Hidden Markov Models

HMMs have been used in several different areas of computational biology such as gene finding, multiple sequence alignments, and profile HMMs used in modeling and alignments of protein families [2], [10], [13], [14].

Formally, a classical HMM λ is characterized by the following parameters [22]:

- A set of N hidden states denoted by $\{Q = \pi_1, \pi_2, \cdots, \pi_N\}$. The state at position t of the observed sequence is denoted by q_t.
- The state transition probabilities given by $N \times N$ matrix A where each element in A denoted by

$a_{ij} = P(q_{t+1} = \pi_j \mid q_t = \pi_i)$ gives the transition

Probability from state π_i to state π_j. Each row of A should add to one (i.e., $\sum_{j=1}^{M} a_{ij} = 1 \; \forall i$).

- An alphabet of M distinct observation symbols per state $\Sigma = \{x_1, x_2, \cdots, x_M\}$. For example, Σ contains *four* letters in case of DNA frequencies: $\{A, C, G, T\}$ or *twenty* letters in case of amino acids.
- The observation emission probability distribution in each of the model states, the probability of emitting the symbol $x_k \in \Sigma$ given that the HMM is in state $\pi_j \in Q$ at position t of the observed sequence.
- The emission probabilities $e_j(x_k)$ form the elements in an $N \times M$ matrix E. The emission probabilities in each state should add to one (i.e., $\sum_{k=1}^{N} e_j(x_k) = 1 \; \forall j$).
- An initial state distribution $\pi_0 = P(q_0 = \pi_i)$, $1 \le i \le N$

We will use the notation $\lambda = (A, E, \pi_0)$ to represent the full set of the HMM parameters.

An important application of HMM framework is in the *statistical pairwise alignment*. This type of model is called pair HMM. The main difference between pair HMMs and classical HMMs lies in the observation of a pair of sequences (the ones to be aligned) instead of a single one.

Pair HMMs provide an alternative formulation of the sequence alignment problem in which alignment generation is directly modeled as a first-order Markov process involving state emissions and transitions [1], [9], [13].

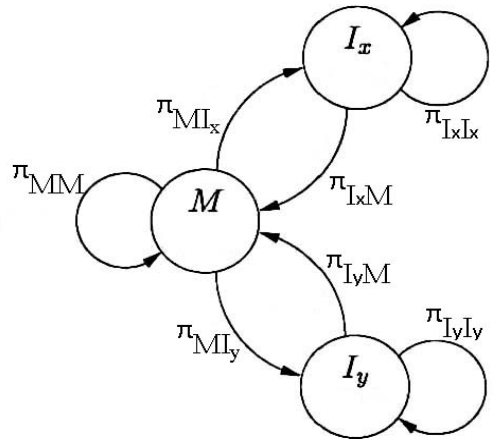

Figure 3: The pair hidden Markov model (HMM) used in statistical sequence alignment. The Viterbi algorithm for this HMM produces the most likely alignment.

Figure(3) shows the basic pair HMM for global sequence alignment between two sequences: X and Y . The model has three states: match state (M), insert state (I_x), and delete state (I_y). The match state generates aligned pairs of bases x_i and y_i . State I_x generates residue x_i aligned to a gap and, similarly, state I_y emits residue y_i aligned to a gap.

The pair HMM model that we adopt in this work is slightly different from the one proposed in [9] in the sense that transition probabilities for the insert and delete states are not equal. Also, the transition probabilities from the match state to either the insert or delete states are not the same. The reason behind that is because these parameters will be estimated from the dynamic programming matrix to approximate a pair HMM representing the actual optimal alignment path as will be explained in Section (3). The most probable path π^*, (i.e., the path of the optimal alignment) through the pair HMM is retrieved by employing the Viterbi algorithm for pair HMMs which calculates the state path with the highest probability. Similar to the standard dynamic programming approach, the Viterbi algorithm determines the most probable state path through two main steps [1], [9], [13]. In the first step the elements of three dynamic programming matrices are iteratively calculated. In the second step the optimal alignment is found by a traceback process.

The Viterbi recurrences are given by the following equations [9]:

$$v^{M}(i,j)=e_{M}(x_{i},y_{j})\max\begin{cases}\pi_{MM}\,v^{M}(i-1,j-1)\\ \pi_{I_{xM}}\,v^{I_x}(i-1,j-1)\\ \pi_{I_{yM}}\,v^{I_y}(i-1,j-1)\end{cases} \tag{2}$$

$$v^{I_x}(i,j)\ =e_{I_x}(x_{i})\max\begin{cases}\pi_{M}\,I_{x}\,v^{M}(i-1,j)\\ \pi_{I_xI_x}\,v^{I_x}(i-1,j)\end{cases} \tag{3}$$

$$v^{I_y}(i,j)=e_{I_y}(y_{i})\max\begin{cases}\pi_{M}\,I_{y}\,v^{M}(i,j-1)\\ \pi_{I_yI_y}\,v^{I_y}(i,j-1)\end{cases} \tag{4}$$

Where $v^{*}(i,j)$ is the probability of emission of the aligned subsequences $x_1,\cdots,x_i$ and $y_1,\cdots,y_j$ by the pair HMM with sub alignment ending with (a) aligned pair x_i and y_j $(v^{*}(i,j)=v^{M}(i,j))$, (b) residue x_i aligned to a gap $(v^{*}(i,j)=v^{x}(i,j))$, and (c) residue y_i aligned to a gap $(v^{*}(i,j)=v^{y}(i,j))$. The match state M has emission probability distribution $e_M(x_i,y_j)$ for emitting an aligned pair x_i,y_i. States I_x and I_y have distributions $e_{Ix}(x_i)$ and $e_{Iy}(y_i)$, respectively, for emitting residue x_i and y_i against a gap.

The probability of the optimal path $P(X,Y,\pi^{*})$ (and ,hence, the optimal global alignment) is found at the termination step of the Viterbi algorithm by taking the maximum of the computed probabilities: $v^{M}(i,j)$, $v^{I_x}(i,j)$, and $v^{I_y}(i,j)$.

New approach to global pairwise alignment in linear space

In this framework, we combine the two approaches to the sequence alignment problem: the standard dynamic programming in linear space, and the statistical pairwise sequence alignment using pair HMMs. The linear space algorithm generates the optimal alignment score only. The traceback step can not be done when the linear space algorithm is used since it requires the whole dynamic programming matrix to be saved. Instead, we propose using Pair HMM model in order to replace the traceback step needed to generate the explicit alignment of the two sequences. In this approach, alignment is generated through *three* main steps as illustrated in Figure (4). First, the linear space alignment algorithm is used to estimate the pair HMM model parameters as will be presented in Section (3). Next, the probability of most likely alignment is obtained using the Viterbi algorithm for pair HMM in its linear space implementation. Finally, since linear space implementation does not allow constructing the actual alignment, a suboptimal alignment is obtained using the trained pair HMM as a *generative* model that generates aligned pair of sequences.

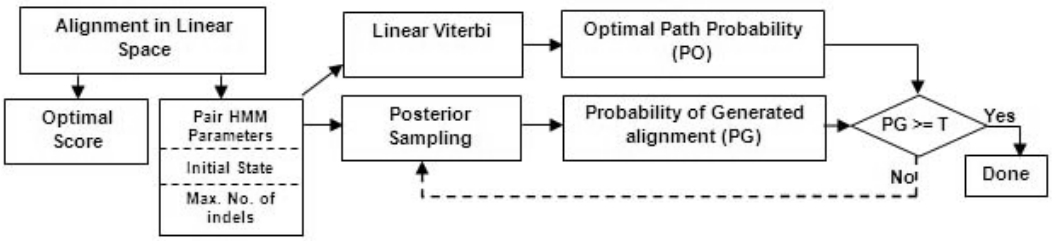

Figure 4: An overview of the proposed new global suboptimal alignment approach combining both standard and statistical alignment algorithms. The dashed arrow represents a repeated step.

The estimated pair HMM starts from the initial state π_0 which is known from the model parameter estimation stage (i.e., the backtracking pointer of the last cell $F(m,n)$ in the dynamic programming matrix). Then, the suboptimal global alignment path is generated by sampling the posterior distribution of the alignments given the learned pair HMM. The alignment generation step is not completely random as it takes into consideration the maximum indel length in the alignment learned during the pair HMM training step.

A major feature of the proposed approach is that the generated global sequence alignment can be *assessed* by comparing the probability of the obtained suboptimal alignment generated by the pair HMM model and the probability of the optimal alignment computed through the Viterbi algorithm implemented in linear space. In case that the relative error between both probabilities is large, then one can repeat the alignment generation process until a satisfactory result is obtained.

Pair HMM Training

The most difficult part in the application of HMMs is the estimation of the model parameters. Usually, the parameters are unknown and should be estimated from the underlying data. In order to be able to find the optimal alignment of two sequences using the Viterbi algorithm for pair HMM, the following pair HMM model parameters must be specified:

- The transition probabilities matrix

$$A = \begin{bmatrix} \pi_{MM} & \pi_{MI_x} & \pi_{MI_y} \\ \pi_{I_xM} & \pi_{I_xI_x} & \pi_{I_xI_y} \\ \pi_{I_yM} & \pi_{I_yI_x} & \pi_{I_yI_y} \end{bmatrix}$$, which is in this case

of size 3×3 since the used model has three states. Note that the transition probabilities $\pi_{I_xI_y}$ and $\pi_{I_yI_x}$ are equal to zero since the adopted alignment model does not allow gaps in one sequence to follow gaps in the other.

- The insert state I_x I_x emission probabilities matrix $E_{I_x} = [P_A \quad P_C \quad P_G \quad P_T]$

- The delete state I_y emission probabilities matrix $E_{I_y} = [P'_A \quad P'_C \quad P'_G \quad P'_T]$.
- The match state M emission probabilities matrix

$E_M = \begin{bmatrix} P_{AA} & P_{AC} & P_{AG} & P_{AT} \\ P_{CA} & P_{CC} & P_{CG} & P_{CT} \\ P_{GA} & P_{GC} & P_{GG} & P_{GT} \\ P_{TA} & P_{TC} & P_{TG} & P_{TT} \end{bmatrix}$, where P_{x_i, y_i} is the probability that the match state M emits residues x_i, y_i .

- The model initial state π_0

Similar to the classical HMMs, there are two main cases when dealing with the pair HMM parameter estimation problem. The first case is when the state path corresponding to the optimal pairwise alignment $\pi^\star = \pi_1, \pi_2, \cdots, \pi_N$ is unknown. The other case is when the optimal alignment path is known. When the optimal path is unknown then the model parameters are usually estimated using Baum-Welch algorithm which is an iterative algorithm. Baum-Welch algorithm is a special version of the general expectation-maximization algorithm [1], [14], [22]. The algorithm starts by a random initial guess for the model parameters $\lambda = (A, E_M, E_{I_x}, E_{I_y}, \pi_0)$ and then iteratively refines the model parameters to improve the total probability of the underlying data. The algorithm stops when the estimated parameters of the current model, $\hat{\lambda}$, change only insignificantly or the maximum number of iterations is exceeded. one major drawback of the Baum-Welch algorithm is that the accuracy of the estimated model parameters and the convergence speed depend on the initial guess of the model parameters [8],[19]. In addition, training pair HMM using the most efficient algorithms for Baum-Welch is too memory-consuming [18].

Motivated by the above facts, our approach to the pair HMM parameter estimation problem is based on estimating the pair HMM model parameters using maximum likelihood estimators. The presented approach is based on the observation that while computing the optimal alignment score in linear space, some of the visited dynamic programming matrix cells are actually those that hold backtracking pointers for the actual optimal or suboptimal alignment paths.

The maximum likelihood estimators for the state transition probabilities a_{ij} and the match state emission probabilities $e_M(x_i, y_j)$ are obtained by counting the number of times each transition or emission is used. If A_{ij} is the number of transitions from state i to j and $E_M(x_i, y_j)$ is the number of times symbols x_i and y_j are emitted from the match state M , then the estimators for both a_{ij} and $e_M(x_i, y_j)$ are given by:

$$a_{ij} = \frac{A_{ij}}{\sum_{q \in Q} A_{iq}} \tag{5}$$

$$e_M(x_i, y_j) = \frac{E_M(x_i, y_j)}{\sum_{\sigma \in \Sigma} E_M(x_i, \sigma)} \tag{6}$$

Similarly, the maximum likelihood estimators for the emission probabilities of the insert and delete states can be obtained by counting the number of times a symbol is

emitted from each state. If $E_{I_x}(x_i)$ and $E_{I_y}(y_i)$ denote the number of times the symbols x_i and y_i are emitted from the insert and delete states respectively, then the maximum likelihood estimators for their respective emission probabilities $e_{I_x}(x_i)$ and $e_{I_y}(y_i)$ are given by:

$$e_{I_x}(x_i) = \frac{E_{I_x}(x_i)}{\sum_{\sigma \in \Sigma} E_{I_x}(\sigma)} \tag{7}$$

$$e_{I_y}(y_i) = \frac{E_{I_y}(y_i)}{\sum_{\sigma \in \Sigma} E_{I_y}(\sigma)} \tag{8}$$

As each new cell in the matrix is entered, there exist *three* possibilities for the state transition. First, if the current cell $F(i,j)$ is derived through its diagonal neighbor $F(i-1,j-1)$ then the current model state is match state (M). Second, if the current state $F(i,j)$ is obtained through its left neighbor $F(i,j-1)$ then the current model state is insert state (I_x). Finally, if the current state $F(i,j)$ is derived through its top neighbor $F(i-1,j)$ then the current model state is delete state (I_y). Once the right transition is determined, equation (5) should be updated to reflect the right count for the underlying transition. Emission probabilities for the various states are computed once that state of the underlying cell is determined. If the state is match (M), the two matched residues in that cell are used to update the count in equation (6). Similarly, if the determined state is insert, then the base from sequence X is used to update the count in equation (7). Finally, if the determined state is delete, then the base from sequence Y is used to update the count in equation (8).

Two important points should be considered when estimating the model parameters while running the standard alignment algorithm in linear space. First, recall that the traceback for the standard alignment algorithm works by building the alignment in reverse starting from the final cell of the dynamic programming matrix $F(m,n)$ and moving back through the predecessor neighbor until the origin of the matrix $F(0,0)$ is reached. Consequently, the occurred state transitions should be considered in *reverse*. In other words, the state transition is counted as it occurs from the current cell $F(i,j)$ to one of its three neighbors which were used in deriving the current cell's score. For example, if $F(i,j)$ is produced through its diagonal neighbor $F(i-1,j-1)$ then the state transition is considered to be a match state M to its diagonal neighbor's state (i.e, $M \rightarrow$ state of $F(i-1,j-1)$).

The other issue is regarding which of the dynamic programming matrix cells should be involved when estimating the model parameters. Using every single cell in the matrix in the computation of the model parameters will lead to overestimation. It is well known that the optimal alignment path exists in a certain band of cells around the main diagonal of the matrix especially if the two sequences are of nearly equal lengths. Also, suboptimal alignment paths do exist in cells that are located in the close vicinity (above or below) of cells associated with the optimal alignment path.

Consequently, we impose a moving window of size w_s on each of the alignment matrix columns as illustrated in Figure (5). The optimal window size is determined empirically as it will be detailed next.

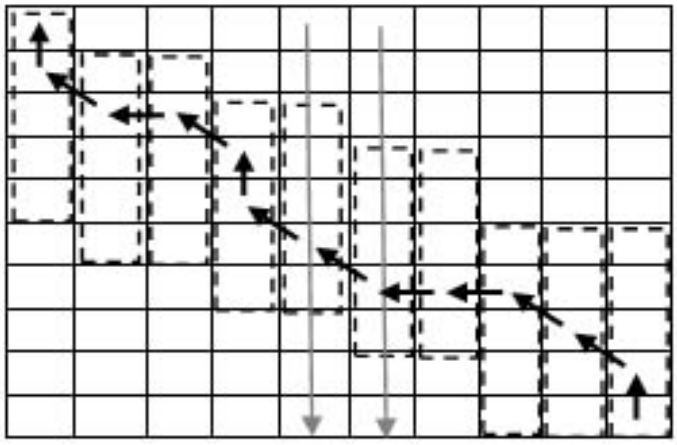

Figure 5: Illustration of the window imposed on each column being processed by the standard algorithm in order to select those cells from which the pair HMM model parameters are estimated.

Training Step Performance

To see the effect of the window size w_s on the estimated model parameters, the full matrix version of the global sequence alignment algorithm is used to produce the actual optimal alignment path for various random pair of DNA sequences with equal length. The true alignment model parameters are computed based on the generated actual optimal alignment path for each sequence pair. In addition, for each sequence pair, the pair HMM model parameters are learned based on the proposed approach for various window sizes.

In both cases, the likelihood estimators given by equations (5-8) are used. The square error between the true model parameters λ computed based on the actual alignment path and the ones obtained using the proposed approach $\hat{\lambda}$ is then computed using:

$$\text{Approximation Error} = \| \lambda - \hat{\lambda} \|_{\infty} \tag{9}$$

Where the norm in equation (9) represents the matrix norm in case of A and E_M and the vector norm in case of E_{I_x}, E_{I_y}.

Figure 6 (a-d) plots the average approximation error given by equation (9) for various window size in the range $[2-N]$, where N is the length of each of the two sequences to be aligned. The average of the approximation error is computed based on several random DNA sequence pairs (generated by **Matlab** *randseq*). Observe that in the case of the transition probabilities matrix A the window size is critical. The error

is large for large window sizes and then decreases as the window size decreases. This suggests that smaller window size will lead to better transition probabilities. For the remaining model parameters, E_M, E_{I_x}, E_{I_y} the estimation error is almost constant regardless of the chosen window size.

In order to study the effect of the window size on the accuracy of the generated global pairwise alignment, the model parameters $\hat{\lambda}$ are estimated using the proposed approach for various window sizes in the range $[2-N]$.

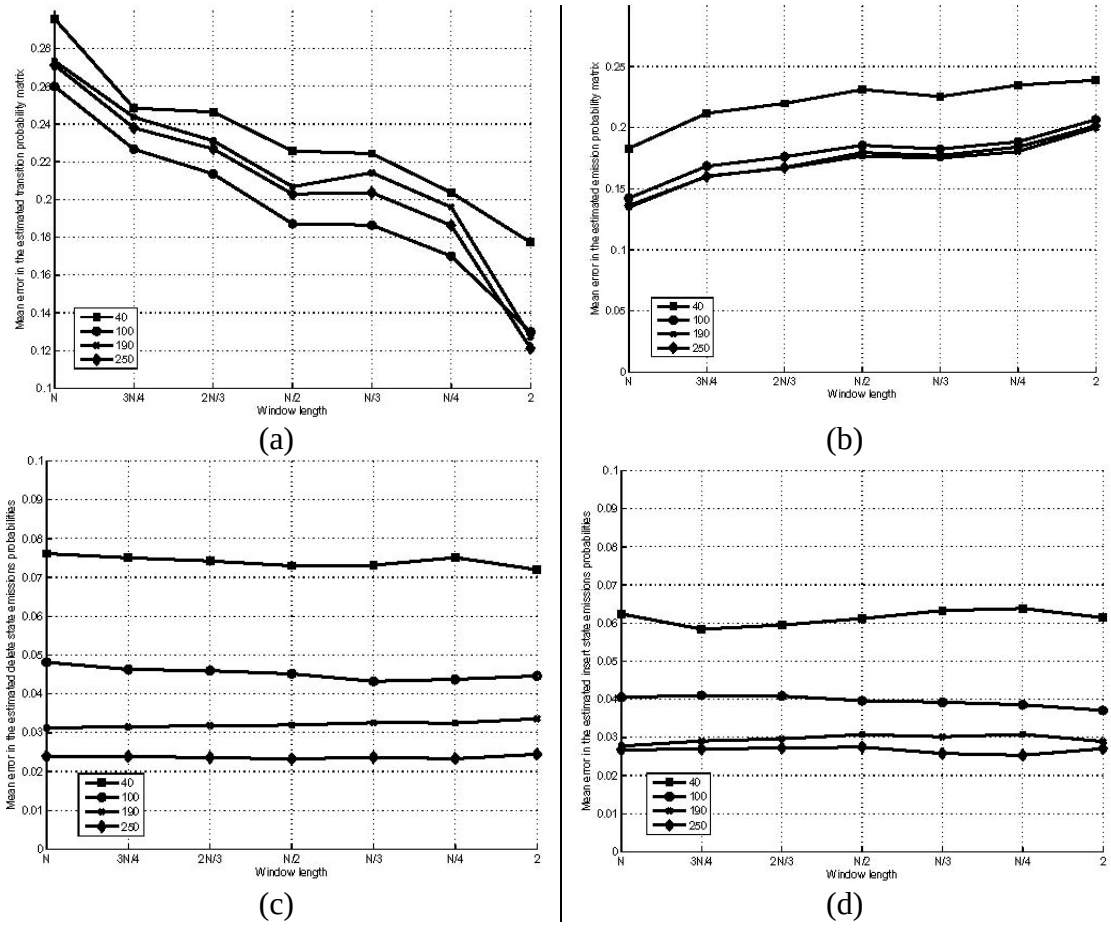

Figure 6: Effect of the window size on the accuracy of the estimated model parameters given by (9): (a) the transition probabilities matrix; (b) the emission probabilities matrix; (c) and (d) the delete and insert states emission probabilities, respectively.

The probability of the optimal global pairwise alignment is obtained using the Viterbi algorithm for pair HMM given in equations (2-4) based on the two types of model parameters: the actual parameters λ computed based on the optimal alignment path generated by the full matrix standard alignment algorithm and the estimated model parameters $\hat{\lambda}$ learned through the proposed approach. The relative alignment

error between the probability of the optimal alignment generated based on both types of the estimated model parameters is then computed using:

$$\text{Alignment error} = \frac{|P(X,Y \mid \lambda) - P(X,Y \mid \hat{\lambda})|}{P(X,Y \mid \lambda)} \tag{10}$$

Where $P(X,Y \mid \lambda)$ and $P(X,Y \mid \hat{\lambda})$ refer to the probability of the most probable alignment obtained based on the two sets of the model parameters: the true parameters and the estimated, respectively.

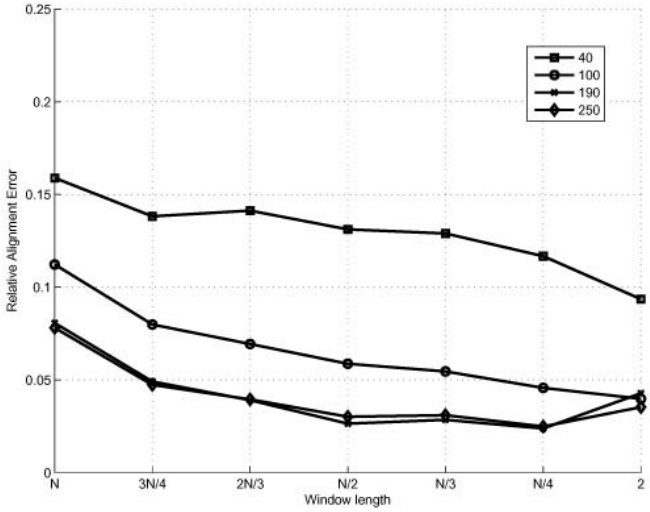

Figure 7: Effect of the window size on the accuracy of the obtained global alignment.

Figure (7) plots the obtained alignment relative error for various random pair of DNA sequences. The relative alignment error decreases as the window size decreases. It is clear that the error behavior is consistent with the observed performance for the estimated pair HMM model parameters in the previous figure.

Alignment Examples

In this section we present global alignment examples generated through the proposed technique. For the purpose of comparing the produced suboptimal alignments based on the proposed method to the optimal alignments produced by both the standard alignment and the Viterbi alignment algorithms, the full matrix version of both the standard alignment algorithm and the Viterbi algorithm is used. Once the pair HMM model parameters are learned based on the proposed method using the optimal window size, the Viterbi algorithm is used to generate the probability of the suboptimal alignment based on the trained pair HMM. Then, the posterior sampling step of the proposed algorithm is used iteratively until a suboptimal alignment is generated with a satisfactory probability.

Figure (8) shows two sets of experiments: the alignments of two sequences of lengths 200; panels (a,b,c).

The optimal alignments generated using the standard dynamic programming algorithm is shown for both experiments in panel (a). The optimal alignment based on the Viterbi algorithm is shown in panel (b). Finally, panel (c) shows the suboptimal alignments generated by "random walk" through the trained pair HMM based on the proposed approach. In this experiment, the posterior sampling step of the proposed algorithm for suboptimal alignment was terminated when the computed relative alignment 34%.

The space complexity of the proposed approach is linear. The time complexity is still linear but depends on number of iteration needed to reach satisfactory alignment accuracy.

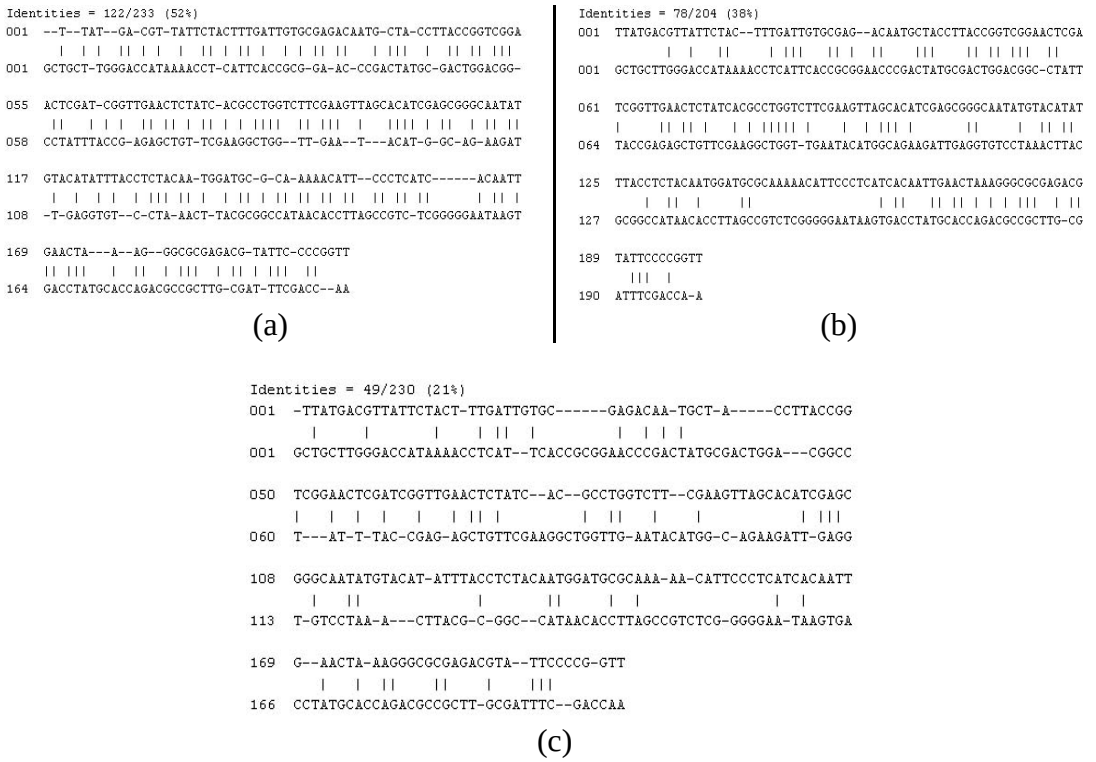

Figure 8: Comparison of alignments for sequences of length 200 produced by: the standard dynamic programming: panel (a) Viterbi algorithm for pair HMM with parameters estimated by the proposed method: panel (b) and a suboptimal alignment produced using the pair HMM as alignment generative model: panel (c).

Conclusions

In this paper, we have proposed a new approach for global pairwise DNA sequence alignment. The proposed approach combines both the standard alignment algorithm

and the probabilistic sequence alignment using pair HMM. We demonstrated through some examples that the proposed method can be applied for the global alignment of large biological sequences. The biological significance of the obtained suboptimal global alignment should be investigated by experts in the field.

Another application of the proposed method is in modeling the global alignment of two sequences by a pair HMM. Given the initial alignment parameters such as the scoring matrix and the gap penalty model, the linear space implementation of the standard alignment algorithm is used to estimate the pair HMM model parameters. The produced pair HMM model can be employed in several bioinformatics applications such as multiple sequence alignment and gene finding.

References

[1] A. Arribas-Gil et al. “Parameter estimation in pair hidden markov models,” Scandinavian Journal of Statistics, Vol. 33, No. 4:651, 2006.

[2] E. Birney. “Hidden Markov models in biological sequence analysis,” IBM J. Res. & Dev., Vol. 45, No 3/4:449–454, 2001.

[3] M. Borodovsky and S. Ekisheva. “Problems and Solutions in Biological Sequence Analysis,” Cambridge University Press, Cambridge UK.,2006.

[4] M. Cameron and H. Williams. Comparing compressed sequences for faster nucleotide blast search. IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 4, No. 3:349–364, 2007.

[5] K. Chao, R. Hardison, and W. Miller. Recent development in linear space alignment methods: A survey. Journal of Computational Biology, Vol. 1:271–291, 1994.

[6] C. Do et al. Probcons: Probabilistic consistency-based multiple sequence alignment. Genome Research, Vol. 15:330–340, 2005.

[7] A. Driga et al. Fastlsa: A fast, linear-space, parallel and sequential algorithm for sequence alignment. Algorithmica, Vol. 45 No. 3:337– 375, 2006.

[8] R. Duda, P. Hart, and D. Stock. Pattern Classification. John Wiley Inc., New york, 2000.

[9] R. Durbin et al. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge UK., 1998.

[10] W. Ewens and G. Grant. Statistical Methods in Bioinformatics: An Introduction. Springer Science Inc., NewYork, 2005. I. Holmes and R. Durbin. Dynamic programming alignment accuracy. Journal of Computational Biology, Vol. 5:493–504, 1998.

[11] N. Johnes and P. Pevzner. An Introduction to Bioinformatics Algorithms. MIT Press, Cambridge MA., 2004.

[12] B. Knudsen and M. Miyamoto. Sequence alignments and pair hidden markov models using evolutionary history. Journal of Molecular Biology, Vol. 333:453–460, 2003.

[13] A. Krogh et al. Hidden markov models in computational biology: Applications to protein modeling. Journal of Molecular Biology, Vol. 235, No. 5:1501–1531, 1994.

[14] W. Majoros, M. Pertea, and S. Salzberg. Efficient implemetationof a generalized pair hidden markov model for comparative gene finding. Bioinformatics, Vol. 21, No. 9:1782–1788, 2005.

[15] [16] I. Meyer and R. Durbin. Comparative ab initio prediction of gene structures using pair HMMs. Bioinformatics, Vol. 18, No. 10:1309–1318, 2002.

[16] I. Meyer and R. Durbin. Gene structure conservation aids similarity based gene prediction. Nucleic Acid Res., Vol. 32, No. 2:776–783, 2004.

[17] I. Miklos and I. Meyer. A linear memory algorithm for baum-welch training. BMC Bioinformatics, Vol. 6:231–237, 2005.

[18] D. Mount. Bioinformatics Sequence and Genome Analysis. Cold Spring Harbor Lab. Press, 2001.

[19] E. Myers and W. Miller. Optimal alignment in linear space. Comp.Appl. in the Biosciences, Vol. 4:11–17, 1988.

[20] S. Needleman and C. Wunsch. A general method applicable to the search of similarites in the amino acid sequence of two proteins. Journal of Molecular Biology, Vol. 48:443–453, 1970.

[21] L. R. Rabiner. A tutorial on hidden markov models and selected Applications in speech recognition. Proc. IEEE, Vol. 77:257–286, 1989.

[22] Batzoglou S. The many faces of sequence alignment. brief bioinform. Briefings in Bioinformatics, Vol. 6:6–22, 2005.

[23] J. Setual and J. Meidanis. Introduction to Computational Molecular Biology. PWS Publishing Comp., Boston, 1997.