ANT COLONY OPTIMIZATION METHODFOR MULTIPLE SEQUENCE ALIGNMENT

LING CHEN[1,2], WEI LIU[1], JUAN CHEN[3]

[1] Department of Computer Science, Yangzhou University, Yangzhou , China
[2]State Key Lab of Novel Software technology, Nanjing University, Nanjing
[3]Yangzhou Institute of Electronic Technology, Yangzhou China
E-MAIL: lchen@yzcn.net, yzliuwei@126.com

Abstract:

Among all the methods for multiple sequence alignment, progressive alignment is the most popular technique because of its simplicity and efficiency. The main drawback of progressive alignment is that the errors occurring in early stages can not be corrected in later stages. In this paper, we propose a novel algorithm called *ProAnt* which combines ant colony optimization and progressive alignment to improve the accuracy of alignment. To avoid the errors occurring in the early stage, the algorithm first calculates the posterior probability of all pairs of characters using ant colony optimization and probabilistic consistency updating. Then the algorithm computes the final alignment using progressive method where the matching score of character pair is replaced by their posterior probability. Experimental results on data from the BAliBASE database show that our algorithm can obtain much more accurate results and higher speed than the other progressive alignment method.

Keywords:

Bioinformatics; Ant colony optimization; Multiple sequences alignment

1. Introduction

Multiple sequence alignment (MSA) problem is by far the most important task in molecular biology and bioinformatics. The simultaneous alignment of many nucleotide or amino acid sequences is a fundamental tool in bioinformatics. It plays an essential role in biological sequence analysis[1], reconstruction of phylogenetic trees [2,3], identification conserved motifs and domains in collections of sequences[4,5], and predicting the secondary and tertiary structure[6,7]. Furthermore, multiple sequence alignment can be a useful technique for studying molecular evolution, RNA folding, gene regulation and protein structure-function relationships. Multiple alignments constitute an extremely powerful means of revealing the constraints imposed by the structure and function on the evolution of a protein family.

The multiple sequences alignment problem requires large amount of computational time. Needleman and Wunsch algorithm [8] is a classical algorithm for pair-wise sequence algorithm. Unfortunately, if Needleman and Wunsch algorithm is generalized for multiple sequences alignment, its space and time cost will be $O(2^N-1)(\prod_{i=1}^{N}|S_i|)$, where, N is the number of sequences and $|S_i|$ is the length of i-th sequence. For practical reasons (in terms of time and memory) this is only possible for sets with less than three sequences.

MSA program [9], an implementation of the Carrillo and Lipman algorithm [10] which finds a way to identify in advance the portion of the hyperspace that does not contribute to the solution and excludes it from computation, can align up to ten closely related sequences. Stoye described a new divide and conquer algorithm DCA [11] that sits on the tops of MSA and extends its capabilities. OMA [12], an iterative implementation of DCA, was presented to speed up the DCA strategy and to decrease its memory requirements. Despite of those improvements, there still exist strong limitations on the number of sequences which can be handled.

Progressive alignments are by far the most widely used heuristic multiple sequence alignment method. This approach has the great advantages of speed and simplicity combined with reasonable sensitivity, even though it is a heuristic that does not guarantee any level of optimization. Among the progressive algorithms, *Clustal-W* [13] is the most popular program based on the improved algorithm presented by Feng and Doolittle [14]. The main drawback of *Clustal-W* is that once two sequences have been aligned, that alignment in the early stage will never be modified even though it conflicts with the other sequences added later. Other progressive alignment methods such as Dialign [15,16] assemble the alignment in a sequence-independent

1-4244-0973-X/07/$25.00 ©2007 IEEE

manner by combining segment pairs in an order dictated by their score, until every residue of all the sequences has been incorporated in the alignment.

Iterative method is another technique for multiple sequences alignment. It refines the alignment through a series of iterations until no more improvements can be made. Iterative methods can be deterministic or stochastic depending on the strategy used to refine the alignment[17-21]. T-Coffee[22], MUSCLE[23] and PROBCONS[24] are the algorithms using integrative technique.

In this paper, we propose an algorithm named *ProAnt* which combines ant colony optimization and progressive alignment to improve the performance of alignment. To avoid the errors occurring in the early stage, we first calculate the posterior probability of all pairs of characters using ant colony optimization and probabilistic consistent updating. Then, we compute the final alignment using posterior probability instead of the matching score of character pairs. Experimental results on data from the BAliBASE database show that our algorithm can obtain much more accurate results and higher speed than the other progressive alignment method.

2. Posterior probability

The progressive alignment consists of three steps. First, it computes the similarities of all possible pair-wise sequence alignment using sequence alignment algorithm. Then, it constructs a guide tree according to the similarities. Finally, it merges the sequences into a multiple alignment along the nodes of a guide tree using the matching score of characters in the sequences.

Since the progressive method is essentially based on pair-wise alignment, errors are easy to occur in the early stage because pair-wise alignment does not take the information of other sequences into account. These errors can not be corrected in later stage. Furthermore, since pair-wise alignment consumes large amount of computation time, progressive method can not obtain high processing speed. Suppose *x, y, z* shown in Figure 1 are three sequences to be aligned. Let x_i , y_j , and z_k be three characters of *x*, *y*, and *z* respectively. In the case only two sequences *x* and *y* be aligned, x_i can be aligned with both y_j and $y_{j'}$ according to their matching score. But in three sequences alignment for *x*, *y* and *z*, when choosing the matching in *y* for x_i, the information of *z* should be taken into account because *z* can supply consistent information for the alignment of *x* and *y*. In Figure 1 we can see that z_k is aligned with x_i and with y_j simultaneously in the three sequences alignment, but it can not be aligned simultaneously with x_i and with $y_{j'}$. Therefore, instead of using the matching score in multiple sequences alignment, we use posterior probability to decide the matching of two characters since the posterior probability take the information supplied by all other sequences into account.

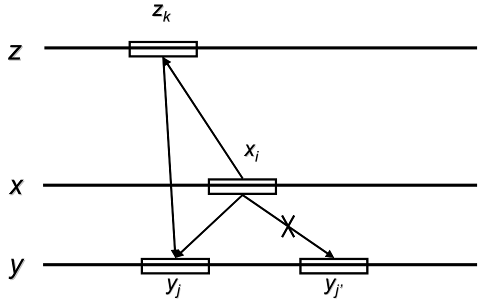

Figure. 1. *Z* supplies consistent information for the alignment of *x* and *y*.

Given a family S= $(S_1,...,S_N)$ of *N* sequences, let $S_k(l)$ be the *l*-th character of S_k . We use $P(S_k(l),S_n(m) \mid S)$ to denote the posterior probability of characters $S_k(l)$ and $S_n(m)$ in the optimal alignment. Since the computation of the posterior probability in *ProAnt* takes the information of all sequences into account, most of the errors occurring in early stage can be avoided. To calculate such posterior probability, the aligned probabilities of all possible character pairs must be computed. In *ProAnt*, we use ant colony optimization and probabilistic consistent updating to obtain the aligned probabilities of all character pairs.

3. Framework of algorithm *ProAnt*

The algorithm *ProAnt* first calculates the aligned probability of all possible character pairs and the similarities of all sequence pairs by the ant colony optimization and probabilistic consistent updating. Then using the aligned probability obtained, the posterior probabilities of all possible character pairs are calculated. After constructing a guide tree, the algorithm merges the sequences into a multiple alignment along the nodes of the guide tree. The framework of *ProAnt* is as follows:

```
Input:   Sequences $S_0,...,S_{N-1}$
Output:  multiple sequence alignment $S'$
 Begin
1.  Using ant colony optimization to calculate the
    similarity of all possible pair-wise sequence alignment
    and the intensity of the   pheromone information
```

between the logic edges between all the pairs of characters from different sequences.

2. Construct a guide tree.
3. Calculate the aligned probabilities $\Pr ob_{k,n}(l,m)$ of all possible character pairs of $[S_k(l), S_n(m)]$ using the pheromone information;
4. Calculate the posterior probabilities $P(S_k(l), S_n(m) \mid S)$ by probabilistic consistency updating using $\Pr ob_{k,n}(l,m)$;
5. Merge sequences into a multiple alignment along the nodes of the guide tree using posterior probability $P(S_k(l), S_n(m) \mid S)$.

End

There are two key points of this algorithm. First is the designing of the posterior probability replacing the matching score in the original progressive method such as *ClustalW*. The posterior probabilities of all possible character pairs are calculated using the aligned probabilitieswhich are obtained by the ant colony optimization and probabilistic consistent updating. Second, the algorithm uses ant colony optimization to calculate the similarity of all possible pair-wise sequence alignment instead of using traditional pair-wise alignment algorithm. This could reduce large amount of computation time.

4. Aligned probability of all character pairs

To obtain the posterior probabilities for the pairs of characters, we first compute their aligned probabilities. All possible pair-wise alignments are searched by ant colony optimization so as to calculate the aligned probabilities of all possible character pairs and the similarities of all the sequences pairs simultaneously .

In the algorithm, artificial ants search for a pair-wise alignment of S_k and S_n by moving on the sequences to choose the matching characters. An ant starts from $S_k(1)$, the first character of S_k, and selects one character of S_n matching with $S_k(1)$. The selected probability of characters in S_n determined by the matching score with $S_k(1)$, deviation of its location from $S_k(1)$ and pheromone trail on the logical edge between the character and $S_k(1)$. In addition, the ant may also select a gap according to a predetermined probability. Next, the ant starts from $S_k(2)$ and selects a character of S_n matching with $S_k(2)$. Similarly, starting from $S_k(3), \ldots, S_k(|S_k|)$, the ant selects their matching characters and finally gets a possible pair-wise alignment. The process is shown in Figure 2.

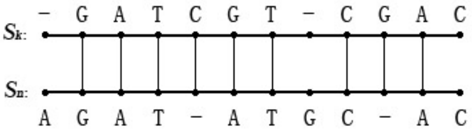

Figure. 2. The process of artificial ant selecting the characters

When the ant selects a character, the path selected should not cross its existing paths since crossover of paths denotes an impossible alignment. It is possible for the ant to select a gap to match a character, this means the gap is inserted into the sequence in the alignment.

Let $P_{k,n}(l,m)$ be the probability for the ant starting from $S_k(l)$ to select the character $S_n(m)$ in S_n . We define $P_{k,n}(l,m)$ as follows:

$$P_{k,n}(l,m) = \frac{phe_{k,n}(l,m) \times a + sim(S_k(l), S_n(m) \times b + dev_{k,n}(l,m) \times c}{\sum_{r=loc(k,l,n)}^{loc(k,l,n)+h} [phe_{k,n}(l,r) \times a + sim(S_k(l), S_n(r)) \times b + dev_{k,n}(l,r) \times c]} \tag{1}$$

Here, $phe_{k,n}(l,m)$ is the pheromone on the logical edge between $S_k(l)$ and $S_n(m)$, $sim(S_k(l), S_n(m))$ is the matching score between $S_k(l)$ and $S_n(m)$, $loc(k,l,n)$ is the start location in S_n when the ant searches for the character matching with $S_k(l)$ within S_n , $dev_{k,n}(l,m)$ is the location deviation between m and l, a>0,b>0,c<0 are the weights of pheromone, matching score and location deviation, and h is the length of the range in S_n for the ant selects the character matching with $S_k(l)$.

By this probability function, the characters in S_n which has higher pheromone on the logical edge connecting with $S_k(l)$, higher matching score with $S_k(l)$ and less deviation to $loc(k,l,n)$ has higher probability to be selected.

When the ant selects a character in S_n to match with $S_k(l)$, it first calculates the selecting probabilities for all characters within a predetermined range in S_n. If the character in S_n which has the largest probability is equal to $S_k(l)$, then it is selected by the ant. Otherwise, the ant selects a character $S_n(m)$ according to the probability $P_{k,n}(l,m)$ or a gap according to a predetermined probability.

When the pair-wise alignment is obtained, its score can be calculated using (2).

$$Score(S_k, S_n) = \sum_{l=1}^{L} sim(S_k^{'}(l), S_n^{'}(l)) \qquad (2)$$

Here, $sim(S_k^{'}(l), S_n^{'}(l))$ is matching score between $S'_k(l)$ and $S'_n(l)$. $S_k^{'}$ and $S_n^{'}$ are the aligned sequences of S_k and S_n. After the score of alignment of S_k and S_n is calculated, we update the pheromone on the paths ants passed according to formula (3).

$$phe_{k,n}(l,m) = phe_{k,n}(l.m) \times (1 - evap) + (alignsum - average) \times evap \qquad (3)$$

Here, *alignsum* is the alignment score of S_k and S_n, *average* is the average score of all alignments by the ant, and *evap* is the evaporation coefficient, $0 \leq evap \leq 1$.

When the algorithm converges, it conserves thephe romone on the paths representing the alignments with *RANGE* highest scores. The aligned probability $\Pr ob_{k,n}(l,m)$ is computed by formula (4) using pherom one in the *RANGE* paths with high scores:

$$\Pr ob_{k,n}(l,m) = \frac{Phe_{k,n}(l,m)}{\sum_{i=0}^{RANGE-1} Phe_{k,n}(l,r_i)} \qquad (4)$$

Here, *RANGE* is a constant, $r_0, \ldots, r_{(RANGE-1)}$ are the indexes of the matching characters of S_n with higher scores. When the algorithm converges, the score of the best solution obtained is the similarity of the sequence pair. The algorithm of ant colony optimization for computing the similarities and the aligned probabilities is described as follows:

```
Input:  Sequences $S_0, S_1, \ldots\ldots, S_{N-1}$
Output: Aligned probabilities $\Pr ob_{k,n}(l,m)$ of
          all possible pairs $[S_k(l), S_n(m)]$ ;
          similarity $sim(S_k(l), S_n(m))$ of all
          pairs of sequences $S_k$ and $S_n$.
Begin
    for $k$=1 to $N$-1 do   // Sequence $S_k$
      for $n$=$k$+1 to $N$ do // Sequence $S_n$
        for cycle=1 to Cyclenum do
          for $r$= 1 to num do      // num ants
            for $l$=1 to $|S_k|$ do
              select a gap according to
              a fixed probability;
              if gap is not be selected
              then
                select a character $S_n(m)$
                according to $P_{k,n}(l,m)$
              end if
            end for $l$ ;
            calculate the score of
            the alignment using(2);
          end for $r$ ;
          update pheromone using(3);
        end for cycle;
        compute $\Pr ob_{k,n}(l,m)$ for all character
        pairs $[S_k(l), S_n(m)]$ in $S_k$ and $S_n$ using
        formula (4);
        conserve RANGE alignments of $S_k$ and $S_n$
        obtained with highest scores, set the the
        highest score of the alignments obtained as
        similarity of $S_k$ and $S_n$;
      end for $n$
    end for $k$
end
```

5. Computing the posterior probability

The aligned probabilities are calculated only based on the alignment of the two sequences the characters located. For multiple sequence alignment, they do not take the information of other sequences into account. But these aligned probabilities can be used to calculate the posterior probabilities by probabilistic consistent updating as follows:

$$P(S_k(l), S_n(m) \mid S) = \frac{1}{N^2} \sum_{\substack{S_i \in S \\ i \neq k,n}} \sum_{j=0}^{|S_i|-1} \Pr ob_{k,i}(l,j).\Pr ob_{i,n}(j,m) \qquad (5)$$

6. Experimental results

We test our algorithm *ProAnt* and progressive alignment algorithm *ClustalW* using the sequences randomly selected from benchmark database BAliBASE 2.0 to compare their alignment qualities and processing speeds.

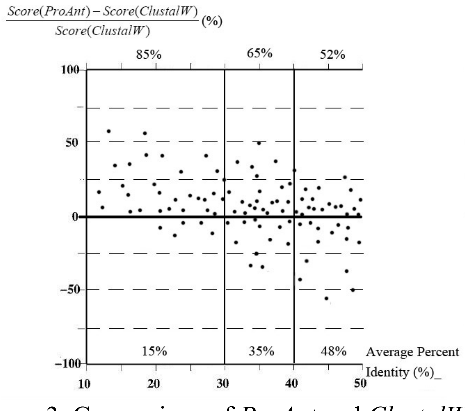

Figure. 3. Comparison of *ProAnt* and *ClustalW*

The qualities of the results by *ProAnt* and *ClustalW* are compared in figure 3 where "●" denotes a result of multiple sequence alignment. The abscissa stands for the similarity of sequences, while the ordinate stands for the value of $\frac{Score(ProAnt)-Score(ClustalW)}{Score(ClustalW)}$, the increased percentage of the score by *ProAnt* over *ClustalW*. It is shown in the figure that the most of the results by *ProAnt* are better than that of *ClustalW*. Especially when the similarity of sequences is between 10% and 30%, 85% of the results by *ProAnt* are better than that of *ClustalW*.

In bioinformatics, if the proportion of the similarity in a set of sequences is below 30%, most of the sequence alignment methods cannot find the correct alignment. Therefore, it is called in the "twilight zone". The experiment results show that *ProAnt* can align not only the sets of similar sequences, but also the sets of in the "twilight zone" efficiently.

Fig.4 shows the comparison of the computation times of our algorithm *ProAnt* with that of Clustal-W. From Fig.4 we can see that *ProAnt* is faster than Clustal-W for sets with different numbers of sequences, especially when the number of the sequences is larger than 5. *ProAnt* costs much less time because it use ant colony optimization to compute the align probabilities of all the character pairs and the similarities of all the sequences pairs simultaneously. It greatly reduces the computational time. *ProAnt* also takes advantage of the strong optimization ability of ant colony algorithm which requires much less time to perform pair wise alignment than traditional methods.

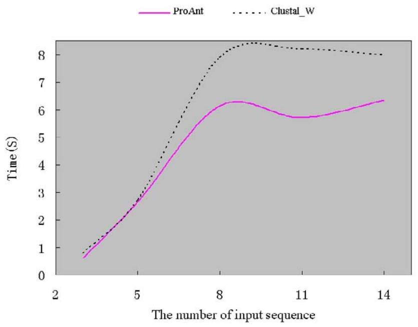

Figure 4. Comparison of the computation time of *ProAnt*with that of *Clustal-W* on sequences sets on different numbers of sequences

7. Conclusions

We present the algorithm *ProAnt* which combines ant colony optimization and progressive alignment to improve the performance of multiple sequences alignment. To avoid the errors occurring in the early stage, we calculate the posterior probability of all pairs of characters and the similarities of all the sequence pairs using ant colony optimization. A guide tree is build using the similarities of the sequence pairs. We construct the multiple sequence alignment along the guide tree using the posterior probability. Experimental results on the BAliBASE database show that the algorithm can obtain much more accurate results and higher speed than the other progressive alignment method.

Acknowledgements

This paper is supported in part by the Chinese National Natural Science Foundation under grant No. 60673060, Chinese National Foundation for Science and Technology Development under contract 2003BA614A-14, and Natural Science Foundation of Jiangsu Province under contract BK2005047.

References

[1] Durbin R et al.: Biological Sequence Analysis. Cambridge University Press. Cambridge, UK (1998).

[2] Felsenstein J: Confidence limits on phylogenies: an approach using the bootstrap. Evolution (1985)39, 783-791.

[3] Aloysius Phillips,1 Daniel Janies, and Ward Wheeler, Review: Multiple Sequence Alignment in Phylogenetic Analysis. Molecular Phylogenetics and Evolution Vol. 16, No. 3, September(2000), pp. 317–330

[4] Luthy R, Xenarios I, Bucher P: Improving the sensitivity of the sequence profile method. Protein Sci. 3(1), (1994), 139-146.

[5] Gribskov M, Luethy R, Eisenberg D: Profile analysis. Meth. Enzymol. 183, (1990) 146-159.

[6] Rost B, Sander C, Schneider R: PHD – an automatic server for protein secondary structure prediction. CABIOS 10, 53-60 (1994).

[7] Jones DT: Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292(2), (1999)195-202.

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

[9] Lipman DJ, Altschul SF, Kececioglu JD: A tool for multiple sequence alignment. Proc.Natl. Acad. Sci. USA 86, (1989) 4412-4415.

[10] Carrillo H, Lipman DJ: The multiple sequence alignment problem in biology.SIAM J. Appl. Math. 48, 1073-1082 (1988).

[11] Stoye J, Moulton V, Dress AW: DCA: an efficient implementation of the divide-andconquer approach to simultaneous multiple sequence alignment. Comput. Appl. Biosci.13(6), (1997) 625-6.

[12] Reinert K, Stoye J, Will T: An iterative method for faster sum-of-pair multiple sequence alignment. Bioinformatics 16(9), (2000) 808-814.

[13] Thompson, JD, Higgins, DG and Gibson, TJ (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. Nucleic Acids Research,(1994), vol.22,No.22.4673-4680.

[14] Feng D-F, Doolittle RF: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol. 25, (1987) 351-360

[15] B. Morgenstern and T. Werner (1997) DIALIGN: Finding local similarities by multiple sequence alignment. Bioinformatics 14, 290-294.

[16] Morgenstern B DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. Bioinformatics, 1 March 1999, vol.15, no. 3, pp. 211-218(8).

[17] Gotoh O: Significant Improvement in Accuracy of Multiple Protein Sequence Alignments by Iterative Refinements as Assessed by Reference to Structural Alignments. J. Mol. Biol. 264(4), (1996) 823-838.

[18] Berger MP, Munson PJ: A novel randomized iterative strategy for aligning multiple protein sequences. Comput. Appl. Biosci. 7, (1991) 479-484.

[19] Notredame C, Higgins DG: SAGA:sequence alignment by genetic algorithm. Nucleic Acids Res. 24, (1996) 1515-1524.

[20] Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science 262,208-214 (1993).

[21] Kim J, Pramanik S, Chung MJ: Multiple Sequence Alignment using Simulated Annealing. Comp. Applic. Biosci. 10(4), (1994) 419-426.

[22] Notredame C, Higgins DG, Heringa J T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol. Sep 8;302(1): (2000) 205-217

[23] Edgar RC. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics. Aug 19;5(1),(2004) 113.

[24] Do, CB, Brudno, M., and Batzoglou, S. 2004. ProbCons: Probabilisticconsistency-based multiple alignment of amino acid sequences. In. Proceedings of the Thirteenth National Conference on Artificial. Intelligence, pp. 703–708.

[25] M. Dorigo, V. Maniezzo, and A. Colorni, The ant system: an autocatalytic optimizing process, Technical Report TR91-016, Politecnico di Milano (1991).

[26] Talbi E.G, Roux O, Fonlupt C, Robillard D.Parallel Ant Colonies for the quadratic assignment problem. Future Generation Computer Systems,17(4),(2001) 441-449.

[27] Ling Chen, Xiao-Hua Xu, Yi-Xin Chen, An Adaptive Ant Colony Clustering Algorithm, Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, August (2004) 26-29.