http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (2): 274-281, 2010
ISSN 1812-5638

Ant Colony with Genetic Algorithm Based on Planar Graph for Multiple Sequence Alignment

[1,2]Xuyu Xiang, [1]Dafan Zhang, [1,2]Jiaohua Qin and [1,2]Yuanyuan Fu
[1]School of Computer and Communication, Hunan University, Changsha, 410082, China
[2]Department of Mathematics and Computer, Hunan City University, Yiyang, 413000, China

Abstract: Multiple Sequence Alignment (MSA), known as NP-complete problem, is among the most important and challenging tasks in computational biology. For multiple sequence alignment, it is difficult to solve this type of problems directly and always results in exponential complexity. In order to effectively solve the MSA problem, in this study, we present a novel algorithm of ant colony with genetic algorithm (ACG) based on the planar graph representation for MSA. Firstly, the planar graph is described a representation for multiple sequences that took every possible aligning result into account by defining the representation of gap insertion, the value of heuristic information in every optional path and scoring rule for the processes of MSA. Secondly, we use an ant colony with genetic algorithm to find the better path that denotes a better aligning result for multidimensional graph. Experimental results show that ACG could bring about a rise in the quality of MSA when compared with standard Clustal algorithm.

Key words: Multiple sequence alignment, planar graph, ant colony algorithm, genetic algorithm, multidimensional graph, local search

INTRODUCTION

Multiple Sequence Alignment (MSA) of nucleotide or amino acid sequences continues to act a very important role in the advancement of understanding in molecular biology. Sequence alignment can be used to determining evolutionary distances between organisms and infer evolutionary history, discover conserved regions among closely related sequences and improve our understanding and prediction of molecular structures (Gusfield, 1997). Therefore, the MSA problem is of great practical interest to biologists.

However, MSA is of very high computational complexity. It is well-known that MSA problems with sum-of-pairs (SP) score can be solved in $O(n^k)$ steps for k sequences of length at most n via Dynamic Programming (DP) (Sankoff, 1972; Carrillo and Lipman, 1988). The size of the MSA problem space increases dramatically with the number of sequences and their lengths. Moreover, it is known that the MSA problem with SP-score is NP-hard (Wang and Jiang, 1994).

Although, DP can guarantee a mathematically optimal alignment, it can only tackle MSA problems with a small number of short sequences. However, this does not exclude the possibility of developing algorithms that produce near optimal MSA in polynomial time. To align multiple sequences within limited resources, a number of align multiple sequences methods have been developed. Many of them use heuristics to find good alignments (Wallace *et al*., 2005; Feng *et al*., 1985; Fogel, 1994).

Though, the progressive alignment methods are faster than DP algorithms, its main disadvantage is the found alignments that may by trapped in local optima, which stems from the greedy nature of this algorithm. This means that if mistakes are made in intermediate alignments, they cannot be corrected later when more sequences are added into the alignment process. Furthermore, there is no objective function to evaluate the performance for multiple sequence alignment.

A number of iterative stochastic approaches have been present to alleviate these problems. For example, evolutionary computation techniques especially Genetic Algorithms (Gas) have been successfully applied to the MSA problem (Notredame and Higgins, 1996; Notredame *et al*., 1997; Chellapilla and Fogel, 1999; Thomsen *et al*., 2002; Wallace *et al*., 2005; Lee *et al*., 2008). It has shown that they can search large solution spaces more efficiently. Nevertheless, when applying GAs to solve complex problems like MSA, they confronted with the conflict between accuracy and speed, GAs sometimes result in an unsatisfactory compromise, characterized by either low solution quality or low

Corresponding Author: Xuyu Xiang, School of Computer and Communication, Hunan University, No. 252, Lushan South Road, Yuelu District, Changsha, 410082, China Tel: 86-731-88821570 Fax: 86-731-88821570

convergence speed. Furthermore, one of the commonest difficulties frequently encountered is premature convergence (Davis, 1981; Fogel, 1994). Recently, GAs with local search have been considered as good alternatives for solving optimization problems (Zhang and Wong, 1997; Jiao and Wang, 2000). The concept of local search is to find a better candidate nearby the current one before move to the next stage of search. In the study of Burke and Smith (2000), Merz and Freisleben (2000) and Lee *et al*. (2003a, b), the genetic algorithm equipped with local search is also referred to as the mimetic algorithm, the genetic local search, the hybrid genetic algorithm, or the cultural algorithm. Studies have shown that the early stages of genetic algorithm in search can fastly convergence to the optimal solution, but the GA is powerless in backtracking, when the solution in a certain extent, GA often do a lot of redundant iterations to make accurate demand solution and reduce efficiency. The ant colony algorithm in the initial search due to lack of information elements, make the slow search speed, but when the pheromone accumulated to a certain intensity, the optimal solution has been rapidly increased the speed of convergence.

In this study, we use a representation for the processes of MSA. We called it planar graph (Chen *et al*., 2009). By this representation, we can consider the every possible aligning result. We also defined the representation of gap insertion, the value of heuristic information in every optional path and scoring rule. We also introduced the ant colony with genetic algorithm (ACG) (Lee *et al*., 2008) to explore and exploit search spaces for MSA. It has both the advantage of genetic algorithm, the ability to find feasible solutions and to avoid premature convergence and that of ant colony algorithm, the ability to search over the subspace and then to move out of local optima.

THE REPRESENTATION OF MULTIPLE SEQUENCES BY USING PLANAR GRAPH

The multiple sequence alignment: Sequence alignment algorithm is to find the best aligning result based on the scoring function, which can calculate the score of any result. The motive is to find the most similar arrangement of several sequences by inserting some gap '-'. Suppose that a family $S = (s_1, s_2, .., s_k)$ of k sequences of various length n_i to n_k is given. Each sequence element represents a character from a given alphabet Σ (for DNA sequences, the alphabet consists of 4 character of nucleotide, i.e., $\Sigma = \{A, T, C, G\}$), an MSA of S is a new k×n matrix of sequences $S' = (s'_1, s'_2, .., s'_k)$ such that all the strings in S' are of equal length and each s'_i is generated from s_i by inserting gaps, where $s'_{ij} \in \Sigma' = \Sigma \cup \{'-'\}$. While performing MSA, we evaluated the quality of the alignment by giving it a numerical score. The SP function is the most popular scoring method for MSA in bioinformatics. The goal of general MSA algorithms is to find out the alignment with the maximum SP score.

The expression of sequences by using planar graph: Multidimensional graphical representation of multiple sequences is the graphic dimension and the number of sequences corresponds. Each dimension represents a sequence. The n-dimensional graph can be expressed as an n-hypercube (Chen *et al*., 2009). The n-dimensional hypercube graph model can be used to describe the Q_n undirected graph, $Q_n = (V, E)$, where V is point set which describe by using all n-bit binary sequence, denoted as $V = \{x_{n-1}x_{n-2}...x_0 | x_i \in \{0, 1\}, i = 0, 1,...,n-1\}$; If and only if the binary strings of points $x = x_{n-1}\, x_{n-2}...x_0$ and $y = y_{n-1}\, y_{n-2}...y_0$ have a different bit, points x and y connected by an edge, E is edge set and is defined as follows:

$$E = \{(x, y) | \sum_{i=0}^{n-1} x_i \oplus y_i = 1,\ x, y \in V\}$$

Figure 1a and b is a three-dimensional and four-dimensional hypercube graph cell.

We describe the hypercube by using the planar graph, which include the vertexes and their adjacency relationship of the hypercube graph. Figure 2a and b show (Chen *et al*., 2009) the planar graph, where all the serial numbers of vertexes are the result of binary-coded

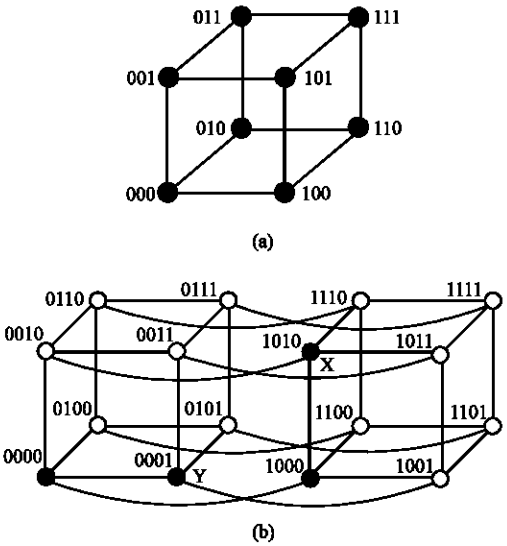

Fig. 1: The hypercube graph cell. (a) The three-dimensional and (b) the four-dimensional

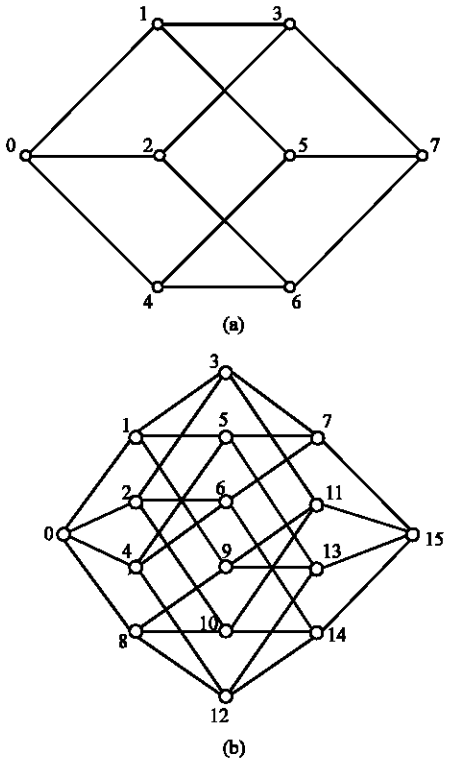

Fig. 2: The planar graph cell. (a) The three-dimensional planar graph and (b) the four-dimensional planar graph

decimal, each dimension of graphics and every edge can be described. The various contacted information in the hypercube graph will be not lost in the planar graph. The information also include all space-diagonal because the diagonal is optional. because in sequence alignment the diagonal direction is optional. At this point it has more easily the alignmen meaning.

THE PROPOSED ANT COLONY WITH GENETIC ALGORITHM FOR MSA

After we advanced the representation for multiple sequences we will introduce the method of use some heuristic algorithm to find the better aligning result in such graph. And we give an example of find the best aligning result by three-dimensional graph and Ant colony with genetic algorithm.

Representation: Given three sequences S_1, S_2, S_3 respectively, the length of m, n, t. With three axes, x-axis, y-axis and z-axis, respectively, said they are out of the points in space corresponding to x, y, z coordinates of the values that they value. And then three-dimensional graphics plane, but that in order to distinguish each point and their coordinates retained. To use k to indicate the direction of the transfer option for each point, k values range: 0, 1, 2, 3, 4, 5, 6, they represent seven different alternative directions. Their corresponding binary code is '000~111'; 0 represent gap '-', 1 represent characters of character set and three representatives of the corresponding x-axis, y-axis, z-axis; such as '101' characters corresponding to said that x-axis is character, y-axis is gap, z-axis is character.

Fitness evaluation: Assume that a scoring matrix C(a, b) is defined to score between any two characters a and b in Σ' with the following characteristics (Chen *et al*., 2009):

$$\begin{aligned} C(a,b) &= +2 \quad \text{if } (a = b \text{ and } a,b \in \Sigma),\\ C(a,b) &= -1 \quad \text{if } (a \neq b \text{ and } a,b \in \Sigma),\\ C(a,b) &= -2 \quad \text{if } (a \in \Sigma, b = '-' \text{ or } b \in \Sigma, a = '-'),\\ C(a,b) &= 0 \quad \text{if } (a = '-', b = '-') \end{aligned}$$

The score between two rows s'_i and s'_j of an alignment S' is defined as follows:

$$C(s'_i, s'_j) = \sum_{p=1}^{n} C(s'_{ip}, s'_{jp}) \quad (1)$$

The score of an alignment S', shown in Eq. 2, is defined by the sum of scores between all pair-wise sequences in S'.

$$C(S') = \sum_{1 \le i < j \le k} C(s'_i, s'_j) \quad (2)$$

The fitness value of an alignment is calculated by Eq. 2. Therefore, the optimization task at hand is defined as a maximization problem, such that matching symbols over columns of an alignment are rewarded and gaps are penalized.

Heuristic information value: For each point, the enabling transfer of the seven selectable direction was inspired by the corresponding information value. According to rating function we can be deduced own evaluation rules (Chen *et al*., 2009). The different types of points in a different direction encountered, are given corresponding to the value of heuristic information. Information value is equal to the sum of its inspired score function value and the absolute value of the minimum score value (negative) and one. Also note that is: these points of the three surface of the cube boundary in the x = m, y = n, z = t have the only three direction to shift, if you select other four direction, you would be beyond the scope of the cube

than the formation of erroneous results. In order to avoid into the four directions, the heuristic information value of the directions were assigned to 0.

State transition rule: For each point, if those inspired information value in these direction is not 0, these directions are the directions of their transition options. We define the integer variable k as the transition direction and its value 0, 1, 2, 3, 4, 5, 6 are corresponding to the seven directions, the probability formula is as follows, at t moment, in the coordinates (i, j, h) of the point of the ants choose the direction of the probability of k forward (Chen *et al*., 2009).

$$P_{ijhk}(t)=\begin{cases}\dfrac{\tau_{ijhk}^{\alpha}M_{ijhk}^{\beta}}{\sum_{s\in allowed^{a}}\tau_{ijhs}^{\alpha}M_{ijhs}^{\beta}} & \text{if } k\in allowed^{a}\\ 0 & \text{else}\end{cases} \quad (3)$$

where, τ_{ijhk} is the k direction of the pheromone, M_{ijhk} represent the direction of the heuristic information value, α and β is the impact factor of these two parameters. $Allowed^{a}$ is the current point the direction of the transfer of a collection of alternative, namely those inspired by the information value of the transfer direction is not 0.

Global pheromone update rule: Each ant has completed a path for the termination condition is to an end (m, n, t), has completed the score after the rules, by definition, a path for each demand points. update the pheromone according to the following formula (Chen *et al*., 2009).

$$\tau_{ijhk}^{new}=(1-\rho)\tau_{ijhk}^{old}+\Delta\tau_{ijhk} \quad (4)$$

$$\Delta\tau_{ijhk}=\sum_{n=1}^{w}\Delta\tau_{ijhk}^{n} \quad (5)$$

$$\Delta\tau_{ijhk}^{n}=\begin{cases}Q\cdot score(n), & \text{if } (i,j)\in route(n)\\ 0, & \text{else}\end{cases} \quad (6)$$

Among them, ρ said the pheromone evaporation coefficient, values range from 0 to 1. Q is a positive number, indicating the distribution of pheromone intensity. The route(n) records the path traversed by the nth ant. The score(n) is the ant paths of the nth ant corresponding to a positive number. It is with this path is proportional to the score values. The w is the number of ants.

The proposed algorithm: Ant colony with genetic algorithm for the specific processes described below:

Step 1: Set the relevant parameters for the ant colony algorithm and genetic algorithm. Cycles NC = 0, set the maximum cycle number NC_{max}. Plan the establishment of three sequences. Each vertex of the graph representative of an optional side of the direction of the initial amount of information set to a constant and initialize the pheromone concentration.

Step 2: Ant colony algorithm iteration number NC = NC+ 1; if the iteration satisfy condition, then the algorithm is ended, otherwise the transfer step 3.

Step 3: Placed in the source node w ant, ants, each based on Eq. 3 state transition probability k and the choice of the next path forward; record the walking path.

Step 4: Placed in the source node w ant, ants, each based on Eq. 3 state transition probability k and the choice of the next path forward; record the walking path.

Step 5: Application of crossover operator and mutation operator to complete the iterative evolution of the genetic algorithm.

Step 6: Iterative process of genetic algorithm to get the optimal solution, according to pheromone update (Eq. 4-6) to update the amount of information on each path.

Step 7: Check to meet the convergence condition, that is, cycles $NC \geq NC_{max}$, if satisfied, then the algorithm is completed and outputs the optimal solution; Otherwise, the return flow step 2.

RESULTS

We download three sequences from different data sources, respectively and we make an experiment with the sample mentioned in Table 1 and we compare our aligning result with the result of ClustalX (Wallace *et al*., 2005) and ACO algorithm in ref (Chen *et al*., 2009).

The CPU and memory size of my computer are 3.0 GHz and 512 MB, respectively. The values of parameters are list in the following:

$\alpha = 0.5$; $\beta = 3$; $\rho = 0.1$; the amount of information the initial value of 2.0; $Nc_{max} = 200$.

We used the customary and acquiescent parameters of ClustalX.

Experiment 1: In the first, we make an experiment with the sample shown in Table 1. Moreover, we compare our aligning result with the result of ClustalX and ACO algorithm.

We can compute the scores of the two results based on the Eq. 2 in Table 2. The scores of ClustalX, ACO algorithm and present proposed algorithm (ACG) are 149, 158 and 167, respectively. Obviously, present result has the highest score. Then, we can count the number of column that has three identical letters. There are 28 columns in the result of the ClustalX; there are 29 columns in AOC; there are 31 columns in our result. Obviously, we can see that present result has the highest columns.

Experiment 2: In addition, we list the sequence segments of three kinds of species and make an experiment. Table 3 shows the original sequences and the aligning results.

We can compute the scores of the two results based on the Eq. 2 in Table 4. The scores of ClustalX, ACO algorithm and our proposed algorithm (ACG) are 345, 348 and 348, respectively. Obviously, present result has a higher score. Then, we can count the number of column that has three identical letters. There are 59 columns in the

Table 1: The original sequences for experiment one and the aligning results

Results	Sequence
Original sequence	
S1	GGCCTCTGCCTAATCACACAGATCTAACAGGATTATTTC
S2	GGCCTCTGCCTTATAACACAAATCTTAACAGGACTATTTC
S3	GGCTTCAGCTTATACACAAATCTTAAACAGGGACTATTC
ClustalX	
S1	GGCCTCTGCCTAATCACACAGATCT--AACAGG-ATTATTTC
S2	GGCCTCTGCCTTATAACACAAATCTT-AACAGG-ACTATTTC
S3	GGCTTCAGCTTA--TACACAAATCTTAAACAGGGACTATTC-
AOC algorithm	
S1	GGCCTCTGCCTAATCACACAGATCT-AA-CAG-GATTATTTC
S2	GGCCTCTGCCTTATAACACAAATCTTAA-CAG-GACTATTTC
S3	GGCTTCAGC-TTATA-CACAAATCTTAAACAGGGACTATTC-
ACG algorithm	
S1	GGCCTCTGCCTAATCACACAGATCT-AA-CAG-GATTATTTC
S2	GGCCTCTGCCTTATAACACAAATCTTAA-CAG-GACTATTTC
S3	GGCTTCAGC-TTAT-ACACAAATCTTAAACAGGGACTATT-C

Table 2: The comparison of the score and same columns for experiment one

Method	Score	The same columns
ACG	167	31
ClustalX	149	28
AOC	158	29

Table 3: The original sequences for experiment two and the aligning results

Results	Sequence
Original sequence	
Lemur	ATGACTTTGCTGAGTGCTGAGGAGAATGCTCATGTCACCTCTCTGTGGGGCAAGGTGG
	ATGTAGAGAAAGTTGGTGGCGAGGCCTTGGGCAG
Gorilla	ATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGA
	ACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
Mouse	ATGGTTGCACCTGACTGATGCTGAGAAGTCTGCTGTCTCTTGCCTGTGGGCAAAGGTG
	AACCCCGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
ClustalX	
Lemur	ATGACTTTGC-TGAGTGCTGAGGAGAATGCTCATGTCACCTCTCTGTGGGGCAAGGTGG
	GATGTAGAGAAAGTTGGTGGCGAGGCCTTGGGCAG-
Gorilla	ATGGT-GCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGT GGGGCAAGGTGA
	ACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
Mouse	ATGGTTGCACCTGACTGATGCTGAGAAGTCTGCTGTCTCTTGCCTGTGGGCAAAGGTG
	AACCCCGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
ACO	
Lemur	ATGACTT-TGCTGAGTGCTGAGGAGAATGCTCATGTCACCTCTCTGTGGGG CAAGGTGG
	ATGTAGAGAAAGTTGGTGGCGAGGCCTTGGGCAG-
Gorilla	ATGG-TGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGC AAGGTGA
	ACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
Mouse	ATGGTTGCACCTGACTGATGCTGAGAAGTCTGCTGTCTCTTGCCTGTGGGCAAAGGTG
	AACCCCGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
ACG	
Lemur	ATGACTT-TGCTGAGTGCTGAGGAGAA-TGCTCATGTCACCTCTCTGTGGGGCAA-GGTG
	GATGTAGAGAAAGTTGGTGGCGAGGCCTTGGGCAG-
Gorilla	ATGG-TGCACCTGACTCCTGAGGAGAAGT-CTGCCGTTACTGCCCTGTGGGGCAA-GGTG
	AACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
Mouse	ATGGTTGCACCTGACTGATGCTGAGAAGT-CTGCTGTCTCTTGCCTGTGGG-CAAAGGT
	GAACCCCGATGAAGTTGGTGGTGAGGCCCTGGGCAGG

Table 4: The comparison of the score and same columns for experiment two

Method	Score	The same columns
ACG	348	62
ClustalX	345	59
AOC	348	60

Table 5: The original sequences for experiment three and the aligning results

Results	Sequence
Original sequence	
Rat β-globin gene	CTCCTGGGCAATATGATTGTGATTGTGTTGGGCCACCACCTGGGCAAGGAATTCACCCC GTCTGCACAGGCTGCCTTCCAGAAGGTGGTAGCTGGAGTGGCCAGTGCCCTTGCTCACAA GTACCACTAA
Lemur (brown) β-globin gene	CTCCTGGGCAACGTGCTGGTGGTTGTGCTGGCTGAACACTTTGGCAATGCATTCAGCC CGGCGGTGCAGGCTGCCTTTCAGAAGGTGGTGGCTGGTGTGGCCAATGCTCTGGCTCACA AGTACCACTGA
Opossum β-hemoglobin β-M gene	ATGCTGGGGAATATCATTGTGATCTGCCTGGCTGAGCACTTTGGCAAGGATTTTACTCCT GAATGTCAGGTTGCTTGGCAGAAGCTCGTGGCTGGAGTTGCCCATGCCCTGGCCCACAA GTACCACTAA
ClustalX	
Rat β-globin gene	CTCCTGGGCAATATGATTGTGATTGTGTTGGGCCACCACCTGGGCAAGGAATTCACCCC GTCTGCACAGGCTGCCTTCCAGAAGGTGGTAGCTGGAGTGGCCAGTGCCCTTGCTCACA AGTACCACTAA
Lemur (brown) β-globin gene	CTCCTGGGCAACGTGCTGGTGGTTGTGCTGGCTGAACACTTTGGCAATGCATTCAGCCC GGCGGTGCAGGCTGCCTTTCAGAAGGTGGTGGCTGGTGTGGCCAATGCTCTGGCTCACA AGTACCACTGA
Opossum β-hemoglobin β-M gene	ATGCTGGGGAATATCATTGTGATCTGCCTGGCTGAGCACTTTGGCAAGGATTTTACTCCT GAATGTCAGGTTGCTTGGCAGAAGCTCGTGGCTGGAGTTGCCCATGCCCTGGCCCACAA GTACCACTAA
ACO	
Rat β-globin gene	CTCCTGGGCAATATGATTGTGAT-TGTGTTGGGCCACCACCTGGGCAAGGAATTCACCCC GTCTGCACAGGCTGCCTTCCAGAAGGTGGTAGCTGGAGTGGCCAGTGCCCTTGCTCACAA GTACCACTAA
Lemur (brown) β-globin gene	CTCCTGGGCAACGTGCTGGTGGT-TGTGCTGGCTGAACACTTTGGCAATGCATTCAGCCC GGCGGTGCAGGCTGCCTTTCAGAAGGTGGTGGCTGGTGTGGCCAATGCTCTGGCTCACAA GTACCACTGA
Opossum β-hemoglobin β-M gene	ATGCTGGGGAATATCATTGTGATCTGC-CTGGCTGAGCACTTTGGCAAGGATTTTACTCC TGAATGTCAGGTTGCTTGGCAGAAGCTCGTGGCTGGAGTTGCCCATGCCCTGGCCCACAA GTACCACTAA
ACG	
Rat β-globin gene	CTCCTGGGCAATATGATTGTGAT-TGTGTTGGGCCACCACCTGGGCAAGGAATTC-ACCC CCGTCTGCACAGGCTGCCTTCCAGAAGGTGGTAGCTGGAGTGGCCAGTGCCCTTGCTCA CAAGTACCACTAA
Lemur (brown) β-globin gene	CTCCTGGGCAACGTGCTGGTGGT-TGTGCTGGCTGAACACTTTGGCAATGCATTC-AGCC CGGCGGTGCAGGCTGCCTTTCAGAAGGTGGTGGCTGGTGTGGCCAATGCTCTGGCTCAC AAGTACCACTGA
Opossum β-hemoglobin β-M gene	ATGCTGGGGAATATCATTGTGATCTGC-CTGGCTGAGCACTTTGGCAA-GGATTTTACTCC TGAATGTCAGGTTGCTTGGCAGAAGCTCGTGGCTGGAGTTGCCCATGCCCTGGCCCAC AAGTACCACTAA

Table 6: The comparison of the score and same columns for experiment three

Method	Score	The same columns
ACG	471	84
ClustalX	471	81
AOC	477	83

result of the ClustalX; there are 60 columns in AOC; there are 62 columns in our result. From the experimental results, we can see that our result has the highest columns.

Experiment 3: We have download the third exon sequences of beta-globin gene of three species and made an experiment. The original sequences and the aligning results are shown in Table 5.

We can compute the scores of the two results based on the Eq. 2 in Table 6. The scores of ClustalX, ACO algorithm and our proposed algorithm (ACG) are 471, 477 and 471, respectively. Present result has a second score. Then, we can count the number of column that has three identical letters. There are 81 columns in the result of the ClustalX; there are 83 columns in AOC; there are 84 columns in our result. From the experimental results, we can see that our result has the highest columns.

DISCUSSION

In present experiments, independent runs of present proposed method, ClustalX and AOC and the statistical

outcome of the optimal fitness in each run are calculated as the results. We measure the score and the number of column with three identical letters (The same column). we compare performance of our proposed method with the two methods through three experiments, the Table 1, 3 and 5 show alignment results of three sequences which were the different source, respectively, Table 2, 4 and 6 show the score value and the same column of alignment results of the three methods in different sources sequences. Table 2 shows the results of the experimental one. From the Table 2, we can see that the score and the same columns of our proposed method are higher than the existing method of ClustalX and the AOC. From Table 4, the second experiment, the score and the same columns of the proposed method are significantly higher than ClustalX, although the score of our proposed method and AOC algorithm is the same, but the same columns of the proposed method are significantly more than AOC.

From Table 6, the third experiment, the score of our proposed method and ClustalX is the same, but the same columns of our proposed method are significantly more than ClustalX. The score and the same columns of our proposed method are significantly higher than ClustalX, although the score of our proposed method is lower than AOC algorithm, but in our method, the same columns are more than AOC. As a whole, our approach is superior to the existing two methods.

CONCLUSION

In this study, we described a new representation for multiple sequences at first. And we use this representation to the MSA. On basis of this representation, we can consider every possible aligning result. We also defined the representation of gap insertion, the value of heuristic information in every optional path and scoring rule. In this kind of multidimensional graph, we use an ant colony with genetic algorithm to find the better path that denotes a better aligning result. In present study, we present the instance of three-dimensional graph and four-dimensional graph and bring forth their multidimensional graph and their planar representation. We advanced a special planar representation to analyze MSA. In the end, we give an example of finding the best aligning result by three-dimensional graph and ant colony with genetic algorithm. Experimental results show that the algorithm can improve the solution quality on MSA benchmarks.

In this study, the proposed ACG algorithm is to enhance the performance of genetic algorithm by incorporating local search, ant colony optimization, for multiple sequence alignment. In the future study, we can enhance the interaction of ants by placing some ants on the end of point. Their tours are from the end to the origin point.

ACKNOWLEDGMENTS

This project is supported by Hunan Provincial National Natural Science Foundation of China (Grant No. 09JJ4033), Scientific Research Fund of Hunan Provincial Education Department (Grant No. 09B019 and 09C210), National Basic Research Program 973 (Grant No. 2007CB310702), National Natural Science Foundation of China (Grant No. 90718008 and 60673155) and Science and Technology Program of Hunan Province (Grant No. 2009FJ3063).

REFERENCES

Burke, E.K. and A.J. Smith, 2000. Hybrid evolutionary techniques for the maintenance scheduling problem. IEEE Trans. Power Syst., 15: 122-128.

Carrillo, H. and D. Lipman, 1988. The multiple sequence alignment problem in biology. J. Applied Math., 48: 1073-1082.

Chellapilla, K. and G.B. Fogel, 1999. Multiple sequence alignment using evolutionary programming. Proceedings of the 1999 Congress on Evolutionary Computation, Jul. 6-9, Washington, DC., USA., pp: 445-452.

Chen, W.B.L., W. Zhu and X. Xiang, 2009. Multiple sequence alignment algorithm based on a dispersion graph and ant colony algorithm. J. Comput. Chem., 30: 2031-2038.

Davis, L., 1981. Handbook of Genetic Algorithms. Van Nostrand Reinhold Company, New York.

Feng, D.F., M.S. Johnson and R.F. Doolittle, 1985. Aligning amino acid sequences: comparison of commonly used methods. J. Mol. Evol., 21: 112-125.

Fogel, D.B., 1994. An introduction to simulated evolutionary optimization. IEEE Trans. Neural Networks, 5: 3-14.

Gusfield, D., 1997. Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge, UK., ISBN: 0521585198.

Jiao, L. and L. Wang, 2000. A novel genetic algorithm based on immunity. IEEE Trans. Syst. Man Cybernet. Part: A Syst. Hum., 30: 552-561.

Lee, Z.J., S.F. Su and C.Y. Lee, 2003a. Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. IEEE Trans. Syst., Man Cybernet. Part B Cybernet., 33: 113-121.

Lee, Z.J., S.F. Su and C.Y. Lee, 2003b. A genetic algorithm with domain knowledge for weapon-target assignment problems. J. Chin. Inst. Eng., 25: 287-295.

Lee, Z.J., S.F. Su, C.C. Chuang and K.H. Liu, 2008. Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. Applied Soft Comput. J., 8: 55-78.

Merz, P. and B. Freisleben, 2000. Fitness landscape analysis and mimetic algorithms for quadratic assignment problem. IEEE Trans. Evol. Comput., 4: 337-352.

Notredame, C. and D.G. Higgins, 1996. SAGA: sequence alignment by genetic algorithm. Nucleic Acids Res., 24: 1515-1524.

Notredame, C., E.A. Brien and D.G. Higgins, 1997. RAGA: RNA sequence alignment by genetic algorithm. Nucleic Acids Res., 25: 4570-4580.

Sankoff, D., 1972. Matching sequence under deletion-insertion constraints. Proc. Natl Acad. Sci. USA., 69: 4-6.

Thomsen, R., G.B. Fogel and T. Krink, 2002. A clustal alignment improver using evolutionary algorithms. Proceedings of the 2002 Congress on Evolutionary Computation, May. 12-17, Honolulu, HI., USA., pp: 121-126.

Wallace, I.M., G. Blackshields and D.G. Higgins, 2005. Multiple sequence alignments. Curr. Opin. Struct. Biol., 15: 261-266.

Wang, L. and T. Jiang, 1994. On the complexity of multiple sequence alignment. J. Comput. Biol., 1: 337-348.

Zhang, C. and A.K.C. Wong, 1997. A genetic algorithm for multiple molecular sequence alignment. Comput. Appl. Biosci., 13: 565-581.