ISSN : 0974 - 7435 Volume 8 Issue 5

BioTechnology
An Indian Journal

FULL PAPER

BTAIJ, 8(5), 2013 [708-713]

A new sequence alignment algorithm based on hybrid intelligence algorithm

Junen Guo*, Huanlong Zhang
Luoyang Institute of Science and Technology, Henan, Luoyan, 471023, (CHINA)
E-mail : gjn_lit@163.com

ABSTRACT

Bioinformatics is the subject of using computer to store, retrieve and analyze biological information. Sequence alignment is a basic problem in Bioinformatics, and its main research work is to develop rapid and effective sequence alignment algorithms. We may discover functional, structural and evolutionary information in biological sequences by sequence comparing. The ant colony optimization genetic algorithm (ACOGA) is an improved algorithm based on the ACO by optimizing its parameters through the GA. In this paper, the ACOGA is applied to sequence alignment in bioinformatics and a novel Hybrid Intelligence Algorithm is proposed. The experiment results indicate that the newly proposed algorithm is effective.

KEYWORDS

Sequence alignment;
Bioinformatics;
Ant colony optimization algorithm;
Genetic algorithm;
Hybrid intelligence algorithm.

INTRODUCTION

Sequence alignment means to compare two or more DNA or protein sequences represented by definite characters through certain algorithms so as to find out their maximal similarity match. After years of research, people have searched out many effective algorithms.

Dot plotting is the basic method in double sequence alignment. In 1970, Gibbs et al.[1] adopted the matrix dot plotting method to seek for similarity match fractions in the sequences. Also in 1970, Needleman and Wunsch[2] proposed the double sequence alignment algorithm on the basis of dynamic planning. In 1981, Smith and Waterman[3] developed the local alignment algorithm on the basis of affine space penalized model, thus enabling the dynamic planning alignment algorithm to seek for those local high similarity fractions in the two sequences. In 1975, according to the divide and conquer strategy, Hirschberg[4] proposed an improved algorithm on the basis of the dynamic planning algorithm. In fact, this is a strategy of trading time for space. In 2000, the FastLSA algorithm proposed by Charter and Schaefer[5] makes a tradeoff between space complexity and running time. In 1978, Dayhoff et al.[6] set up the PAM substitution matrix after studying the evolution of protein sequences. In 1992, Henikof et al.[7] introduced the later widely adopted BLOSUM substitution matrix into the alignment algorithm. Currently, the FastA and the BLAST are two famous search algorithms, both of which are based on local similarities. The FastA algorithm was put forward by Pearson and Lipman[8] in 1985, which can search in DNA and protein databases. The BLAST algorithm was put proposed by Altschul et al. in 1990 and has become the most popular tool in

database searching. Henceforth, any improved BLAST algorithm[10] allows the insertion of blank slots while the UniBLAST[11] tool kit can filter those meaningless sequences from the abundant sequences obtained from searching.

In recent years, studies which apply the optimal algorithm to sequence alignment have been increasing gradually. Mologni and Shinoda proposed an algorithm that uses the genetic algorithm (GA) to conduct double sequence alignment. In their framework, a batch of sequence alignment results are grouped into an initial population, in which better alignment results will be selected to be duplicated, and the crossover and mutation operators are used to generate the next generation of population, and finally the optimal alignment results will be obtained by convergence through the evolution of multiple generations. In 1997, Cedric, Emmet and Desmond[12] proposed a genetic algorithm (GA) for multisequence alignment. The ant colony optimization algorithm (ACO) is a new kind of simulated evolutionary algorithm, the use of which to solve the sequence alignment problem is an attempt of the optimal algorithm in this field. Reference[13] applied the ACO to the sequence alignment in bioinformatics.

In this study, the ACOGA is applied to sequence alignment, thus fixing up the defects of the ACO's local optimal results. The experiment results show that our algorithm is more effective than that proposed in reference[13].

THE SEQUENCE ALIGNMENT ISSUE IN BIOINFORMATICS

The main idea of the sequence alignment (SA) method is as follows: for a sequence S, |S| is the length of sequence S, and Si represents the ith character in sequence S. $_i$:S:$_j$ indicates that the sequence between i and j is a subsequence of sequence S. The characters in S are determined by a certain finite character set Ω (for example, DNA is determined by A, T, C and G; a single-letter amino acid is determined by 20 kinds of letters, etc.). The variation of genetic sequences in mutation includes substitution, insertion, and deletion. We use "-" to represent the blank slots generated by insertion and deletion. For $x,y \in \Omega \cup \{-\}$, $f(x,y)$ is defined as a scoring function to represent the score obtained when x is compared with y. Equation (1) is one of its forms:

$$f(x,y)=\begin{cases}2, x=y\in\Omega\\ 1, x\neq y\in\Omega\\ -1, x='-' \text{ or } y='-'\end{cases} \quad (1)$$

A sequence alignment A between S and T is represented by the one-to-one correspondence between the characters in the sequences, in which '$|S'|=|T'|$ and a$|S'|S'$, $|T'|T'$. The deletion of the spaces with result in S and T. the score of A is:

$$Score(A)=\sum_{i=1}^{|S'|}p(S'_i,T'_j) \quad (2)$$

The alignment which results in the maximal value is the optimal alignment.

THE DEFICIENCIES OF THE ACOSA ALGORITHM

The ant colony optimization (ACO) algorithm simulates the behavioral feature of ants when they are autonomously searching for the shortest path in the course of carrying food. When improved, the ACO algorithm can be applied to different fields. The ant colony optimization sequence alignment (ACOSA) algorithm designed in reference[13] is easy to fall into the local optimal result. Therefore, a new kind of ACOGASA algorithm is designed in this study to overcome this deficiency. The basic idea of this algorithm is as follows: first, transform the ACO so as to make it apply to the sequence alignment (SA) algorithm, that is, the new kind of ACOSA algorithm; then, use the genetic algorithm (GA) to optimize a group of the parameters of the ACO algorithm so as to expand the search space of the ACO algorithm, thus making the final algorithm not be easy to fall into the local optimal result and consequently increasing the possibility of converging to the global optimal result. The experiment results show that this method is more effective than the algorithm designed in reference[13].

ALGORITHM DESIGN

The design of the new kind of ACOSA

For sequences S=CAGGA and T=CGGTTA, the matrix shown in Figure 1 is set up by imitating the dynamic planning method. The ant starts from the top left

FULL PAPER

corner and chooses a path to reach the bottom right corner, thus forming an alignment. We designate that when a cell is moved horizontally or vertically, it means that a blank slot is inserted in the corresponding sequences; and that when a celled is moved diagonally, it means the match between the characters corresponding to the newly reached positions.

	-	C	G	G	T	T	A
-							
C							
A							
G							
G							
A							

Figure 1 : The walked path of a single ant

The path in Figure 1 represents the following alignment results:

Sequence S': CAGG—A

Sequence T': C–GGTTA

According to the computations on the basis of equation (1) and equation (2), the score of the alignment results is 5. In all the paths found, the alignment which scores the highest is the optimal alignment. Sequence alignment has certain requirement for the number of blank slots inserted. Therefore, the design of the ACOSA algorithm is different from the traveling salesman problem (TSP).

In the newly designed ACOSA algorithm, ant z (z=1, 2, …m) starts from the top left corner of the matrix, and according to the probability defined by equation (3), moves to the next position until it reaches the bottom right corner:

$$\mathbf{p_{ijk}^{z}(t) = \frac{(\tau_{ijk}(t))^{\alpha}(\eta_{ijk}(t))^{\beta}}{\sum_{k=0}^{2}(\tau_{ijk}(t))^{\alpha}(\eta_{ijk}(t))^{\beta}}} \quad \textbf{(3)}$$

$$\mathbf{\eta_{ijk}(t) = (f(x+y)+2)/5} \quad \textbf{(4)}$$

$\tau_{ijk}(t)$ represents the pheromone density along the path R_{ijk} in the kth direction at position (i,j) in Figure 1 at moment t. In the equations, i=0, 1, 2, …|S|, j=0, 1, 2, …|T|, and k=1, 2, 3, representing the rightward, rightdownward and downward directions. The initial moment is set as $\tau_{ijk}(0) = \tau_0$ (τ_0 is a constant).

$\eta_{ijk}(t)$ represents a scale operator, whose value is between 0 and 1, which is in direct proportion to the matching score. If the ant's present position is (i,j), if the direction k=1 or k=3 is chosen, then the alignment score will be -1 according to equation (1); if the direction k=2 is chosen, then the alignment score will be 2 or 1. However, when the ACO algorithm is used to solve the TSP problem, the η_{ij} factor is a scale factor, whose value is between 0 and 1, which is in inverse proportion to the distance between cities. Hence, the $\eta_{ijk}(t)$ here should also have this sense, namely should be a scale operator, whose value is between 0 and 1, which is in direct proportion to the matching score. On the basis of this point, the score value should be mapped into a value which can satisfy this condition. The simplest way is to add a positive value to the score so as to change the score into a positive value in the first place, and then divide it by a larger integer so as to transform it into a value which satisfies the condition. This value can be adjusted accordingly depending on the differences in equation (1). α and β are the weight values allocated to pheromone and heuristics information, thus embodying the degree of their influence on decision. The two values can be adjusted reasonably in experiments.The selection strategy adopted by this study is as follows: first, set q_0 "(0,1) $q_0 \in (0,1)$, when the ant is choosing the path, a random number p is generated from (0,1); when $p \le q_0$, ant z chooses the direction k whose value is the largest in $p_{ijk}^{z}(t)$ (k=1,2,3); when $p > q_0$, ant z randomly chooses one of the three directions and walks along.

When all the ants reach the bottom right corner of the matrix by different paths, a group of alignment results will be obtained, thus finishing one loop of searching for the optimal path. At this moment, global updating is necessitated for the pheromone density on each path, because new pheromone will be added in and old pheromone will volatilize. Set $\rho \in (0,1)$ as the volatilization coefficient, then $1-\rho$ reflects the volatilization degree of the pheromone. The updating equations are (5) and (6):

$$\mathbf{\tau_{ijk}(t+n) = \rho \times \tau_{ijk}(t) + \Delta\tau_{ijk}} \quad \textbf{(5)}$$

$$\mathbf{\Delta\tau_{ijk} = \sum_{z=1}^{m}\Delta\tau_{ijk}^{z}} \quad \textbf{(6)}$$

$\Delta\tau_{ijk}$ the pheromone increment for path R_{ijk} during

this loop. The initial moment $\Delta\tau_{ijk} = 0$. $\Delta\tau_{ijk}^{z}$ represents the pheromone amount left by the zth ant along the path, as shown in equation (7):

$$\Delta\tau_{ijk}^{z} = \begin{cases} Q \times A_z, \text{if ant z has passed } R_{ijk} \\ 0, \text{otherwise} \end{cases} \quad (7)$$

$$A_z = \frac{Score^{z} + MaxScore}{2 \times MaxScore} \quad (8)$$

Q is a constant; A_z correlates with the score of the sequence alignment represented by the path walked by ant z. In the TSP, L_k is used to represent the path length, which is always larger than zero. However, the score of sequence alignment can be negative. Therefore, the score should be mapped into a positive value before allocating it to A_z. The higher the alignment score, the larger the . The mapping method adopted in this study is similar to that of , as shown in equation (8), in which represents the total score that ant z has obtained from its paths during this round; *MaxScore* represents the possible maximal score obtained from computations from equation (1) and equation (2).

Because the ants choose different paths, the lengths of the paths they covered may not be equal, namely, they may not reach the bottom right corner of the matrix at the same time. Therefore, a flag whose initial value is 0 is added to each ant, and this flag will be set to 1 when the ant reaches the bottom right corner. When all the ants reach the bottom right corner and after the global updating of the pheromone has been finished, all the ants will be relocated to the top left corner of the matrix, set the flag to 0, and start the next round of loop. The algorithm will terminate when the maximal evolution generation number has been reached or when the optimal result remains unchanged for ten consecutive generations.

In order to quicken the search, the optimal result of each loop is recorded and then compared with other optimal results obtained from other loops so as to obtain the final optimal result.

The design of the ACOGASA algorithm

Pilat and White[15] adopted the genetic algorithm (GA) to optimize three the parameters ,and of the ACO algorithm and conduct a simulation verification in the traveling salesman problem (TSP). This study adopted the GA algorithm to optimize four parameters (,,and) of the ACO algorithm to expand the solution space and overcome the deficiency of the ACO in being easy to fall into the local optimal result, and then applied it to the sequence alignment in bioinformatics. The experiment results show that the effect of this ACOGASA algorithm in sequence alignment is relatively remarkable.

The ACOGASA algorithm can be described as follows:

Step 1: according to the parameter combinations (α, β, ρ and q_0) of the ACO algorithm, generate the initial population, and meanwhile set ;

Step 2: decode every individual in the population, obtain the values of ,,and respectively through computations, and obtain the fitness value for every individual in the population according to the following steps;

① Initiation: search loop number NC=1, $\Delta\tau_{ijk} = 0$, set the flags of all the ants flag=false;

② For ant z=1, 2, …, m, carry out the following operations:

If the ant does not reach the bottom right corner of the matrix, then according to the generated random variable to determine to use equation (3) or randomly choose a path, move the ant to a new position; if the ant does reach the bottom right corner of the matrix, then set flag=true;

③ When all the ants have reached the bottom right corner of the matrix, compute each ant's pheromone increment according to equation (6) and equation (7);

④ Conduct updating to the pheromone on the paths according to equation (5), record the optimal result obtained from the present loop, NC=NC+1; put all the ants to the top left corner of the matrix, set flag=false, $\Delta\tau_{ijk} = 0$;

⑤ If or the optimal result remains unchanged for ten consecutive loops, then turn to ②; otherwise record the optimal result obtained from the present loop as the individual's fitness value;

Step 3: Select those individuals for the next generation according to the roulette rules determined by the individuals' fitness values;

Step 4: Conduct crossover operation according to probability p_c ;

FULL PAPER

Step 5: Conduct mutation operation according to probability p_m;

Step 6: If the termination conditions are not met, then turn to Step 2; otherwise, go on to Step 7;

Step 7: Use the individual whose fitness value is optimal in the output population as the problem's satisfying or optimal result, and decode and obtain the values of α, β, ρ and q_0 respectively corresponding to the optimal result through computations, and then obtain the optimal sequence alignment and the score.

EXPERIMENT RESULTS

From the GenBank database we chose two DNA sequences NM_003533 and NM_003536 whose lengths were 477 and 472 characters to conduct alignment and use the BLAST matrix to mark the score. From the PDB database we chose two amino acid sequences AAX46609 and XP_533117 whose lengths were both 492 characters to conduct alignment and use the PAM-250 matrix to mark the score. Then we adopted NC=100 (iteration number), m=30 (total number of ants), $\tau_0 = 1.0$, $\alpha = 2.0$, $\beta = 1.0$, $\rho = 0.8$, $q_0 = 0.2$ and Q=100 as the parameters of the ACOSA; we adopted $NC_{\max} = 100$ (number of ant searching times), m=10 (total number of ants), , Q=100, population size = 4, $p_c = 0.8$, $p_m = 0.08$, MaxGeneration (genetic generation number)=100 as the parameters of the ACOGASA algorithm; and then we respectively used the ACO algorithm in reference[13] and the approach proposed in this study to conduct ten rounds of experiments on the two DNA sequences and the two amino acid sequences, and obtained the minimum score, the average score, and the maximum score, as shown in TABLE 1.

From TABLE 1 we can clearly see that the ACOGASA algorithm is better than the ACOSA algorithm. Meanwhile, when the optimal results were obtained for the DNA and amino acid sequences, we decoded the optimal individual obtained from the ACOGASA algorithm to get the parameters of the ACO algorithm. After analyzing these parameters, we obtained the approximate value ranges of them, as shown in TABLE 2.

TABLE 2 : The value ranges of the parameters (α, β, ρ and q_0)

α	β	ρ	q_0
2~5	3~5	0.1~0.6	0~0.3

TABLE 1 : The minimum score, the average score, and the maximum score obtained from the ten experiments on the DNA and amino acid sequence alignments

Sequences aligned	Sequence alignment algorithm	Minimum value	Average value	Maximum value
NM_003533 and	ACOSA	1642	1672.3	1689
NM_003536	ACOGASA	1684	1691.2	1693
AAX46609 and	ACOSA	2261	2271.1	2273
XP_533117	ACOGASA	2267	2276.6	2278

CONCLUSION

The present study applied the ACOGA algorithm to the sequence alignment (SA) in bioinformatics and obtained desirable SA effects, thus providing a new approach to the SA in bioinformatics. In fact, so long as we use different scoring functions and scoring matrices, we can obtain different experiment results, and the ACOGA algorithm can also be applied to the SA in RNA. And work in this aspect will be content of our further research in the future.

REFERENCES

[1] A.J.Gibbs, G.A.Mcintyre; The diagram: a method for comparing sequence.Biochem., **16**, 1-11 **(1970)**.

[2] B.Needleman Saul, D.Wunsch Christian; A general method applicable to the search for similarities in the amino acid sequence of two proteins.Mo1.Biol., **48**, 443-453 **(1970)**.

[3] T.F.Smith, S. M.Waterman; Identification of common molecular subsequence.Mol.Biol., **147**, 195-197 **(1981)**.

[4] D.S.Hirschberg; A linear space algorithm for com-

puting maximal common subsequence.Communications of the ACM, **18(16)**, 341-343 **(1975)**.

[5] K.Charter, J.Schaefer; Sequence alignment using FastLSA.International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences.METMBS, 239-245 **(2000)**.

[6] M.O.Dayhoff, R.M.Schwartz, B.C.Orcutt; A model of evolutionary change in protein.Atlas of Protein Sequence and Structure, 345-352 **(1978)**.

[7] S.Henikof, G.Henikof Joria; Amino acid substitution matrices from protein blocks.Proc.Natl Acad.Sci., USA, **89**, 10915-10919 **(1992)**.

[8] D.J.Lipman, W.R.Pearson; Rapid and sensitive protein similarity searches.Science, **227**, 1435-1441 **(1985)**.

[9] S.F.Altschul, W.Gish, W.Miller, E.W.Myers, D.J.Lipman; Basic local alignment search tool.Journal of Molecular Biology, **215**, 403-410 **(1990)**.

[10] S.F.Altschul, T.L.Madden, A.A.Schafer, et al.; Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.Nucleic Acids Research, **17(25)**, 3389-3402 **(1997)**.

[11] Y.Zhou, G.M.Huang, L.P.Wei; UniBLAST: A system to filter, cluster, and display BLAST results and assign unique gene annotation.Bioinformatics, **9(18)**, 1268-1269 **(2002)**.

[12] Notredamel Cedric, A.O'brien Emmet, G.Higgins Desmond; RAGA: RNA sequence alignment by genetic algorithm.Nucleic Acids Research, **22(25)**, 4570-4580 **(1997)**.

[13] D.Liang, H.W.Huo; An adaptive ant colony optimization algorithm and its application to sequence alignment.Computer Simulation, **1(22)**, 100-106 **(2005)**.

[14] M.Dorigo, V.Maniezzo, A.Colorni; Ant system: optimization by a colony cooperating agents.IEEE Trans.on Systems, Man, and Cybernetics-Part B: Cybernetics, **26(1)**, 29-41 **(1996)**.

[15] M.L.Pilat, T.White; Using genetic algorithms to optimize ACS-TSP.Proceedings of the 3rd International Workshop on Ant Algorithms/ ANTS2002.Lecture Notes in Computer Science, **2463**, 282-287 **(2002)**.