Genetics and Molecular Research

Multiple sequence alignment based on combining genetic algorithm with chaotic sequences

C. Gao, B. Wang, C.J. Zhou and Q. Zhang

Key Laboratory of Advanced Design and Intelligent Computing, Dalian university, Ministry of Education, Dalian, China

Corresponding author: Q. Zhang
E-mail: zhangq26@126.com

Genet. Mol. Res. 15 (2): gmr.15028788
Received May 16, 2016
Accepted June 3, 2016
Published June 24, 2016
DOI http://dx.doi.org/10.4238/gmr.15028788

ABSTRACT. In bioinformatics, sequence alignment is one of the most common problems. Multiple sequence alignment is an NP (nondeterministic polynomial time) problem, which requires further study and exploration. The chaos optimization algorithm is a type of chaos theory, and a procedure for combining the genetic algorithm (GA), which uses ergodicity, and inherent randomness of chaotic iteration. It is an efficient method to solve the basic premature phenomenon of the GA. Applying the Logistic map to the GA and using chaotic sequences to carry out the chaotic perturbation can improve the convergence of the basic GA. In addition, the random tournament selection and optimal preservation strategy are used in the GA. Experimental evidence indicates good results for this process.

Key words: Multiple sequence alignment; Genetic algorithm; Chaotic sequences; Logistic map

INTRODUCTION

Bioinformatics (Fan, 2012) is a discipline that uses computer technology to study the laws of a biological system. With the successful implementation of the human genome project and the initiation of a variety of post-genome projects, a large quantity of biological molecular data is emerging. Among these data sets exists a vast amount of hidden biological knowledge, and sequence alignment (He, 2010) forms the basis of analyzing this molecular data. Through sequence alignment analysis, we can assess the similarity between multiple sequences, to ascertain homology. Therefore, research of MSA methodologies in bioinformatics has very important theoretical and practical significance.

Sequence alignment is one of the core components of bioinformatics, and the basic method to analyze various sequences. Presently, there are many algorithms for sequence alignment, but most are based on the basic idea of the dynamic programming algorithm. The sequence alignment algorithm is divided into the double sequence alignment algorithm (Wu and Chen, 2008) and MSA algorithm (Zou et al., 2010). MSA suffers from the same problems as double sequence alignment. To solve this problem, many approximate algorithms (Song et al., 2005) and heuristic algorithms have been proposed (Altschul and Lipman, 1989; Yuan and Li, 2013). The most typical algorithms for MSA are the dynamic programming algorithm (Hu, 1987; Wang, 1993; Gan et al., 1994), central star alignment algorithm (Zou et al., 2009), and iterative comparison algorithm (Zhang et al., 2005).

With the problem put forward, many algorithms have been proposed to solve MSA. Because of the increasing number of sequences available, the complexity of the algorithm must also increase rapidly, producing a greater requirement for computer system resources, rendering these algorithms impractical. In MSA, a local optimal solution is difficult to achieve. The GA has the advantages of fast computation speed, comprehensive search ability, and the selection of suitable mutation probability, which can effectively avoid the problem of local optimization (Chen et al., 1996; Ma et al., 2010).

At present, there are many GAs involved in the MSA (Zhang and Achawanantakun, 2010) using the evolutionary computation system written in JAVA, resulting in the realization of this process. It was reported that the GA in the reserved region successfully avoided the premature phenomenon and improved the MSA of short sequences. However, there was no significant improvement for datasets with long sequences. (Pramanik and Setua, 2014) used a steady state GA with a new form of chromosome representation. The results were tested using a BAliBASE benchmark dataset, which showed that this solution did offer better results. (Orobitg et al., 2013) investigated a combined scoring function, which was capable of obtaining a good approximation of the biological quality of the alignment. The algorithm used the information obtained by the different quality scores to improve accuracy. Results showed that the combined score was able to evaluate alignments better than the isolated scores. He and Zhou (2010) proposed a hierarchical GA, with the use of different genetic operators in the low layer GA, and achieved the overall convergence speed, and avoided the effect of local convergence. (Fan, 2012) presented three ways to improve the GA. The first was to put forward some intelligent operators, which considered the characteristics of the biological sequences. The second was the use of the variance of fitness value to adjust the probability of intelligent mutation and random mutation to maintain the diversity of the population. The last one was to use the cycles of evolution to void the large setting of the generation. (Deblasio and Kececioglu, 2015) developed a greedy approximation algorithm that found near optimal

sets of size, given an optimal solution of size. They found that the coefficients for the estimator performed well in practice. (Mirarab et al., 2014) introduced a new and highly scalable algorithm, PASTA, for MSA estimation. They presented a study on biological and simulated data with more than 200,000 sequences, proving that the algorithm could produce accurate alignments and was able to analyze much larger datasets.

According to the present research situation, an improved GA is proposed, which combines chaotic sequences and the GA to solve MSA. The GA has some limitations, such as premature convergence, and lack of ability to produce the optimal individual, and even the local optimal solution, as well as other defects. Chaos is characterized by randomness, ergodicity, and regularity. We used Logistic map (Lu et al., 1990; Fan and Zhang, 2009) to generate chaotic sequences, and disturb the crossover and mutation in the process of MSA, so that the improved GA can effectively avoid the problem of premature convergence. By comparing to other algorithms, we found that the algorithm is effective and outperforms other algorithms in most cases.

MATERIAL AND METHODS

Multiple sequence alignment

Sequence alignment consists of DNA, RNA, and protein sequence alignment. The DNA sequence character set is $\sum = \{A, C, G, T\}$. The RNA sequence character set is $\sum = \{A, C, G, U\}$. For protein sequence characters, of which there are 20, each character represents an amino acid.

Suppose that there are N sequences (Li et al., 2004):

$$S_1 = S_{11}S_{12}\cdots S_{1l_1}$$
$$S_2 = S_{21}S_{22}\cdots S_{2l_2}$$
$$\vdots$$
$$S_N = S_{N1}S_{N2}\cdots S_{Nl_N}$$

Among them, $s_{ij} \in \sum$, $i = 1,2,...,N$, $j = 1,2,...,l_i$, (l_i is the length of each sequence). $\sum$ represents a set of amino acids or nucleotides, but it does not contain characters "-". Using "-" indicates the phenomenon of insertion or deletion of residues in the process of genetic evolution. A comparison between S_1, S_2, ...S_N is D, where D is a two-dimensional character matrix (Li et al., 2004):

$$D = \begin{bmatrix} b_{11}b_{12}\cdots b_{1L} \\ b_{21}b_{22}\cdots b_{2L} \\ \vdots \\ b_{N1}b_{N2}\cdots b_{NL} \end{bmatrix}$$

Where $b_{ij} \in \sum'$, $i = 1,2,... N$, $j = 1,2,... L$, and $\max\{l_i\} \le L \le \sum_{i=1}^{N} l_i$. D satisfies three conditions (He and Zhou, 2010): 1) If you remove the "-" character in each row in a 2D character matrix, you will get the original sequence; 2) The number of rows for the 2D character matrix is equal to the number of sequences; 3) In any column, same character "-" is not allowed.

The results of multiple sequence alignment and double sequence alignment are the same. The results can be used to measure the similarity between them. The same residues are assigned to the positive points, and different residues are assigned to the penalty. Finally, the total score is calculated. To evaluate the quality of the alignment, we need to define an appropriate objective function (Li et al., 2004):

$$V_{\text{AlignScore}} = \sum_{n=1}^{L} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} V_{Score}\left(b_i, b_j\right) \qquad \text{(Equation 1)}$$

Here, N is the number of rows, and L is the number of columns. $V_{Score}(b_i, b_j)$, is the sequence alignment score between S_i and S_j.

GA

The GA is a global search algorithm, which is based on the natural selection and evolution of the biological community, with inherent parallelism, and thus can solve large-scale problems. Different from the traditional optimization search method, the GA is especially suitable for dealing with nonlinear and complex problems, and is effective in solving the NP problem.

The GA can obtain the global optimal solution by selection, crossover, and mutation. As a search algorithm, the GA has unique advantages. However, as an optimization algorithm, the GA has some disadvantages. First, the GA is a random search algorithm, in which the initial population generation, crossover, and mutation are all completed random. Therefore, the calculation efficiency of this algorithm is relatively low. Second, some experimental data show that the GA can often appear as the premature convergence phenomenon. Finally, the random of the GA cause some individuals to produce premature convergence.

Improved GA

From the above analysis, the traditional GA can result in premature convergence and low efficiency; the most important reason for this is that the algorithm uses randomness. Therefore, this study makes use of chaos theory to improve the operation of crossover and mutation. Controlling the crossover and mutation operation of the genetic algorithm with chaotic sequences can improve the efficiency of the genetic operation. At the same time, it also makes some improvements in the generation of the parent population and the selection operation.

Selection operation

This procedure uses the random tournament selection method to make a selection. The tournament method removes a certain number of individuals from the parent population each time, and then chooses one of the best as a child. These operations are repeated until the size of the sub population reaches the prescribed population size.

Chaotic crossover

Traditional genetic crossover is based on a given rule from the parent to choose two individuals randomly, and then follows the crossover probability p_c to perform crossover operations, randomly select the intersection point, and exchange the corresponding parts.

The crossover operation is one of the most important operators in the GA. There are two random factors in the cross operation. The first is the frequency of the crossover operation, and the crossover probability, p_c. Beyond that, the choice of the cross point, is also randomly selected.

From the point of view of mathematical form, a one-dimensional Logistic map is a very simple chaotic map; previously, several ecologists used the simple difference equation to describe the changes in a population. This system has a very complex dynamic behavior, and it is now widely used in science and technology. Its mathematical expression is (Yao et al., 2001) as follows:

$$W_{n+1} = \mu \times W_n \times (1 - W_n) \qquad \text{(Equation 2)}$$

Where $\mu \in [0,4]$ is e parameter of the Logistic map. According to the relevant research, when $W \in [0,1]$, the Logistic map presents a complete state of chaos. Random selection of an initial value W_0 in the Logistic map generates a non-periodic, non-convergence, better pseudo random sequence. When μ = 4, the sequences $\{W_n\}$ generated by equation 2 (Yao et al., 2001) are called chaotic sequences.

The basic idea that we used for the chaotic sequences to control crossover is as follows: randomly take an initial value W_0, and put W_0 into the Logistic map to generate an iterative sequence W_{n+1}, the value of this sequence W_{n+1} is changed back and forth between [0,1]. Then put W_{n+1} as a random control variable. When the value of W_{n+1} is greater than the given crossover probability p_c, the cross operation is carried out, and on the contrary it is not carried out.

Chaotic mutation

In the same way, we can control the mutation operation through the chaotic sequences.

The idea of using a Logistic map to generate the chaotic sequences to control the mutation operation is to take a random number W'_0, generating sequence W'_{n+1} by the Logistic map. Then put W'_{n+1} as a random control variable; when the value of W'_{n+1} is greater than the given variation probability p_m, mutation operation is carried out, otherwise it is not carried out.

We used the generated chaotic sequences to control the probability of crossover, and

the probability of controlling the mutation operation. The advantage of this method is that the process of the improved GA makes full use of the chaotic randomness and ergodicity. Therefore, crossover and mutation have inherent regularity, and avoid the traditional GA, which is purely random. We developed a combination of the GA and chaotic sequences, which operated better together.

SPS score

We usually use SPS (sum of pairs score) scores to evaluate the merits of multiple sequence alignment algorithms.

Suppose we have a test alignment with N sequences consisting of M columns. The i-th column in the alignment can be represented by $A_{1i}\, A_{2i} \ldots A_{Ni}$. For each pair of residues A_{ji} and A_{ki}, we define $P_{j\,ki} = 1$ if residues A_{ji} and A_{ki} are aligned with each other in the reference alignment, otherwise $P_{j\,ki} = 0$. The score of S_i for the i-th column is defined as:

$$S_i = \sum_{j=1}^{N} \sum_{k=1, j \neq k}^{N} P_{j\,ki} \qquad \text{(Equation 3)}$$

The SPS for the alignment is:

$$SPS = \frac{\sum_{i=1}^{M} S_i}{\sum_{i=1}^{M_r} S_{ri}} \qquad \text{(Equation 4)}$$

where M_r is the number of columns in the reference alignment and S_{ri} is the score S_i for the i-th column in the reference alignment.

The implement algorithm

According to the traditional GA and the improved method, we propose a MSA method based on chaotic sequences. A flow chart of the improved GA based on this method is shown in Figure 1, with the following specific steps:

Step 1: Initializing parameters and population; Step 2: The fitness of a MSA is calculated by the sum of all pairs of characters at each column in the alignment and the dynamic programming algorithm is used. According to equation 1, each individual fitness value is calculated, and the maximum fitness value of the individual is selected as the parent; Step 3: Using the method of random tournament selection operation, select the individual with a higher degree of adaptation from the population to carry out the next step operation; Step 4: Using the method of section 2.3.1, the single point crossover operation is controlled by the chaotic sequences; Step 5: Using the method of section 2.3.2, the mutation operation is controlled by the chaotic sequences; Step 6: The original sequences are processed by the

improved genetic manipulation, and new sequences are obtained. Re-calculating the fitness value of the new sequences, and sorting the largest and the smallest values. Then, make the individual fitness value maximum instead of the fitness value being the smallest; Step 7: Judging termination condition. First, we must carry out the update operation of the genetic algebra counter $gen = gen + 1$. When $gen < MAXGEN$, it does not comply with the conditions of termination and return to Step 3; when $gen > MAXGEN$, it is consistent with the termination conditions, proceed to the next step; Step 8: Outputting the optimal alignment score.

From the specific process, we identified that the improved GA is a very good solution for sequence alignment.

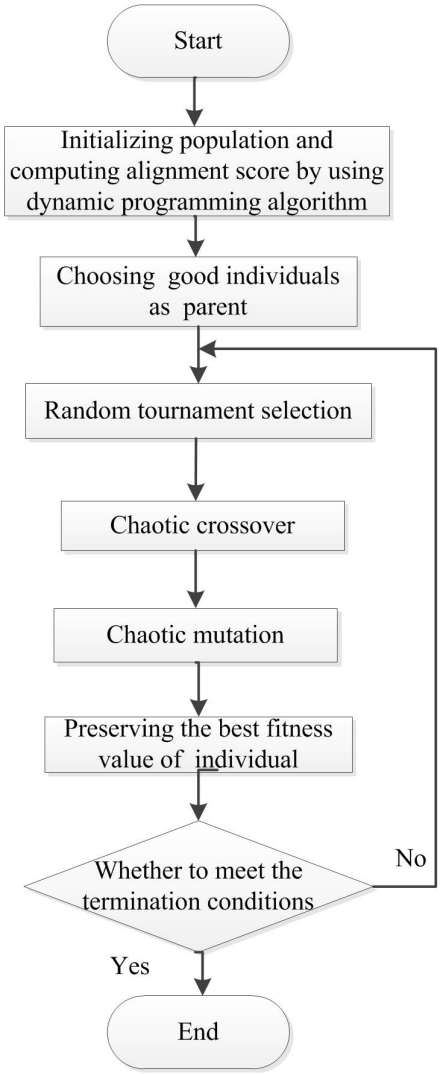

Figure 1. Flow chart of improved GA.

RESULTS

Our algorithm was run in the Windows 7 operating system (64 bit) using R2012a MATLAB. The processor of the system was an Intel(R)Core(TM)i3-4160 CPU @3.60GHz with 4.00G RAM.

In this algorithm, two parameters needed to be configured. The first was the GA parameter. The crossover probability was 0.8, and a single point crossover was selected; the mutation probability was 0.01, using a single point mutation; the population size was 50 and the iteration number was 2000. The second parameter was the alignment score and scoring rules as shown in equation 3. Σ represented the character set of the input sequence and contained no spaces.

$$M(x,y)=\begin{cases}6 & \text{if } x, y \in \sum \text{ and } x = y,\\ 1 & \text{if } x, y \in \sum \text{ and } x \neq y,\\ 0 & \text{if } x \in \sum \text{ and y is a gap opening,}\\ 1 & \text{if } x \in \sum \text{ and y is a gap extension.}\end{cases} \quad \text{(Equation 5)}$$

In sequence alignment, the DNA sequences, RNA sequences, and protein sequences are the most common types of sequence alignment. In this experiment, we calculated the experimental results of RNA sequence alignment.

Experimental data

We obtained real data from an online database (http://www.ncbi.nlm.nih.gov/nuccore) to test our algorithm. The data we employed belongs to the BAliBASE 2.1 database. BAliBASE is a benchmark database for MSA. Each set of data contains five sequences of equal length. Detailed information of the data set is shown in Table 1 (Project Report, CSE848, 2010). The detailed parameters of the GA are shown in Table 2 (Project Report, CSE848, 2010).

Table 1. Sequence data.

Sequence	Sequence length	Sequence name
TAR	57	L28864.1_329-385 M93259.1_9532-9588 AF443088.1_8897-8953 AF196710.1_461-517 AJ286133.1_8742-8798
IRES_PICO	252	AF230973.1_399-650 D00627.1_394-645 AF524867.1_393-644 AY186745.1_373-624 AJ295195.1_354-605

Experimental results

To demonstrate the feasibility of our proposed method, we used two different RNA sequences for analysis. For the first short data set, it took less than five minutes to run the program to the 2000th generation. For the longer sequences in the second group, the time

spent was longer. We put the score as the main measure of MSA. Experimental results of this study and the experimental results of (Project Report, CSE848, 2010) are shown in Table 3.

Table 2. GA parameters.

Parameter	TAR	IRES_PICO
Population size	50	100
Crossover operator	Chaotic crossover	
Crossover probability	0.8	
Mutation operator	Chaotic mutation	
Mutation probability	0.01	

Table 3. Experimental results.

Sequence	Best result	Project report, CSE848	Traditional GA	This paper
TAR	Alignment score	3,206	3,096	3,223
	SPS	0.93	0.984	0.984
IRES_PICO	Alignment score	13,457	12,425	13,676
	SPS	0.86	0.972	0.996

In this paper, we present two analytical methods to verify the contention presented in this thesis. The first method was to compare our results with the results of (Project Report, CSE848, 2010). The second method was to compare the results with the best results from the traditional the GA.

In the TAR MSA: for the first method, the best score in (Project Report, CSE848, 2010) was 3206, while the best score obtained herein was 3223. Therefore, our methods outperformed those of (Project Report, CSE848, 2010). From the perspective of SPS, our experimental results were better than those of (Project Report, CSE848, 2010). For the second method, using the traditional GA, the best score was 3096; therefore, the results of the proposed method were better than those of the traditional GA. In the IRES_PICO MSA, we can see that the results of the improved GA were better than those of the other two methods tested. Therefore, the improved GA is able to obtain good results through the addition of chaotic sequences. From the results of SPS, we can conclude that the improved algorithm is better than (Project Report, CSE848, 2010).

DISCUSSION

Problems associated with MSA were solved by combining chaotic sequences and the GA. To improve the search ability and efficiency of the GA, we used the Logistic map to generate the chaotic sequences. Beyond that, random tournament selection and an optimal preservation strategy are used in the selection strategy. Our experimental results show that GA based on chaotic sequences can be superior to other methods. Therefore, the improved algorithm is an effective and feasible method for MSA.

The algorithm has the advantage of a fast convergence rate and can be performed faster. However, there was one limitation to this method. The main disadvantage of this algorithm is poor stability. In the future, we will strive to solve the stability problem of the algorithm.

ACKNOWLEDGMENTS

Research supported by the National Natural Science Foundation of China (#61425002, #61402066, #61402067, #31370778, and #61370005), the Basic Research Program of the Key Lab in Liaoning Province Educational Department (#LZ2014049 and #LZ2015004), the Project Supported by Natural Science Foundation of Liaoning Province (#2014020132), the Project Supported by Scientific Research Fund of Liaoning Provincial Education (#L2014499), and by the Program for Liaoning Key Lab of Intelligent Information Processing and Network Technology in University.

REFERENCES

Altschul SF and Lipman DJ (1989). Trees, stars and multiple biological sequence alignment. *SIAM J. Appl. Math.* 49: 197-209. http://dx.doi.org/10.1137/0149012

Chen GL, Wang XF and Zhuang ZQ (1996). Genetic algorithm and its application. People's Posts and Telecommunications Press, Beijing.

DeBlasio D and Kececioglu J (2015). Parameter advising for multiple sequence alignment. *BMC Bioinformatics* 16 (Suppl 2): A3.

Fan H (2012). The Research of Sequence Alignment Method based on Genetic Algorithm. Master Thesis, Hunan University, Chang Sha.

Fan JL and Zhang XF (2009). Piecewise Logistic chaotic map and its performance analysis. *ACTA Electronica Sin.* 4: 720-725.

Gan YA, Tian F and Li WZ (1994). Operations research, Tsinghua University Press, Beijing.

He XM (2010). Research of Multiple Sequence Alignment Based on Genetic Algorithm. Master Thesis, Inner Mongolia Agricultural University, Hohhot.

He XM and Zhou GB (2010). Research of multiple sequence alignment based on genetic algorithm. *J. Inner Mongolia Agric. Univ.* 3: 267-271.

Hu YQ (1987). Research Foundation and Application, Harbin Institute of Technology Press, Harbin.

Li SZ, Mo ZS and Zhang X (2004). Multiple sequence alignment based on immune genetic algorithm. *J. Wuhan Univ.* 05: 537-541.

Lu K, Sun J, Ouyang R, et al. (1990). Chaotic dynamics. Shanghai Translation Press, Shanghai.

Ma Y and Yun WX (2010). The research progress of genetic algorithm in the large warehouse system. Optoelectronics and Image Processing, International Conference, Haiko, 616-619.

Mirarab S, Nguyen N and Warnow T (2014). PASTA: Ultra-Large Multiple Sequence Alignment. Springer International Publishing, Switzerland, 177-191.

Orobitg M, Cores F, Guirado F and Notredame C (2013). Improving multiple sequence alignment biological accuracy through genetic algorithms. *Springer Science Business Media* 65: 1076-1088.

Pramanik S and Setua SK (2014). A steady state genetic algorithm for multiple sequence alignment. International Conference on Advances in Computing, Communications and Informatics, 1095-1099.

Project report of CSE848 (2010). Project Report. CSE848, Michigan.

Song B, Chen GL and Yan C (2005). Parallel approximation algorithm for multiple sequence alignment problem. *J. Univ. Sci. Technol. China* 5: 78-86.

Wang YX (1993). Planning and network of operations research, Tsinghua University Press, Beijing.

Wu DM and Chen J (2008). Research on algorithm of pairwise alignment. *Comp. Engineering Appl.* 44: 48-50.

Yao JF, Mei C and Peng XQ (2001). Chaos genetic algorithm and its application. *Syst. Eng.* 1: 70-74.

Yuan JJ and Li Y (2013). Accelerating database sequence comparison on CUDA platform. *Intelligent Comp. Appl.* 2: 44-49.

Zhang M, Fang WW and Zhang JH (2005). Algorithm based on iteratively progressive multiple sequence alignment. *Comp. Engineering* 17: 32-33+61.

Zou Q, Guo MZ, Wang XK and Zhang TT (2009). An Algorithm for DNA multiple sequence alignment based on center star method and keyword tree. *Acta Electronica Sin.* 8: 1746-1750.

Zou Q, Guo MZ, Han YP and Li WB (2010). Development of multiple sequence alignment algorithms. *China J. Bioinformatics* 4: 311-315.