International Journal of Engineering & Technology, 7 (4.19) (2018) 751-756

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Optimal Pair DNA Sequence Alignment based on Matching Regions and Multi-Zone Genetic Algorithm

Sara Q. Abedulridha*[1], Eman S. Al-Shamery[2]

[1]*MSc. Student, S/W Dept., IT College. University of Babylon, Iraq*
[2]*Assistant prof., Dr., S/W Dept., IT College. University of Babylon, Iraq*
**Corresponding Author Email: sara_alghanimi@yahoo.com*

Abstract

DNA sequence alignment is an important and challenging task in Bioinformatics, which is used for finding the optimal arrangement between two sequences. In this paper, two methods are proposed in two stages to solve the pairwise sequence alignment problem. The first method is Matching Regions(MR) concerns on splitting the DNA into regions with adaptive interleaving windows to isolate the DNA tape into matched and non-matched regions. Additionally, a Multi-Zone Genetic Algorithm (MZGA) is proposed as an improved method in the second stage. It consists of segmenting a large non-matched region into smaller search space. Then, the MZGA is implemented in parallel to save time. Genetic Algorithm can be applied as an optimization toolto produce multiple solutions. Furthermore, the improvement focuses on the enhancement of Simple GA operators. In the selection, the population is divided into three Zones according to the fitness score. A new crossover approach is proposed depending on cut-points and location of gaps. The proposed method guarantees that the value of fitness tends to improvement or convergence in each successive generation. Thus, the offspring of populations will have better fitness value. The system has been applied to the real-world dataset of DNA with variable lengths which are ranged from 66 bases up to 26037 bases. As a result, the proposed technique satisfied the best alignment score of the DNA sequences. Finally, it is worth mentioning that the proposed system proved to be generalizable.

Keywords: *Bioinformatics, DNA Sequence Alignment, Matching regions, Multi-Zone Genetic Algorithm, Parallel processing.*

1. Introduction

Nowadays DNA sequence alignment is an active research area within the field of Bioinformatics. DNA sequence alignment plays the primary part in finding the related sequences which are beneficial to extract the information/knowledge of structure properties, function and evolutionary history of a DNA sequence[1].

DNA Sequence Alignment means two or a more sequence of DNA and RNA nucleotide, or amino acids of a protein are compared to align their bases[2].

In the field of sequence alignment, there are three fundamental issues. The first is pairwise sequence alignment (PSA) that is lining up two sequences to get the biggest level of identity for similarity. The second is how to compare a specific sequence with the database of sequences[3]. Third, is Multiple Sequence Alignment (MSA) which align up more than two sequences.

Recently, DNA sequencing technologies produce A huge amount of biological sequence[4]. However, the computational algorithms for the sequence alignment do not appear to be adequately efficient in comparison with the size of the sequence data.

For the first problem, the highest scoring alignment is regularly found through the dynamic programming algorithm (DP), for example, Needleman- Wunsch (1970) is used for global alignment, and Smithe Waterman (1981) is used for local alignment. Regardless,they verified to achieve optimal alignment with PSA tasks, it needs a major quantity of memory and cannot be used for alignment of very long sequences or extended for multiple sequences alignments. For the long sequence, more than 100,000 DP build a scoring matrix containing more than 100,000*100,000 elements [5], which may overflow the memory of a computer. Moreover, For MSA Dynamic programming suffer from high dimensionality, where the number of matrix dimensions is equal to the number of sequences. The complexity of time and space will become O(n^d), where n is the sequence length and d are the numbers of sequences. It is a large combinatorial problem (NP-hard)[6]. The computational effort becomes prohibitive[7][8]. For the second problem, BLAST [9] and FASTA [10]are the most common heuristic algorithm used to align the pairing. BLAST looks for similarities in local alignment by comparing individual residues within the two sequences[11]. while FASTA searches for similarities in local alignment by comparing sequence patterns or words. Regardless, they're quicker strategies, they cannot guarantee the optimal global finding. They concern more about computational efficiency than alignment accuracy. For the last issue, different algorithms like iterative, progressive, and consistency-based techniques[12], [13]are used, for example, ClustalW [14], MUSCLE [15], and T-Coffee package[16]. Their methods take time and space complexity for long sequences.

Other than these above-mentioned DNA sequence alignment methods, An evolutionary algorithms Genetic algorithm was enhanced to deal with the Pair DNA Sequence Alignment and then extended to solve multiple DNA Sequence Alignment[13][17][18]. GA used for implementing an Iterative method.

GA different from other optimization methods. It requires just a fitness function, rather no specific algorithm to solve a given problem. The fitness function is the cost function given using different weights for various kinds of matching symbols and assigning gap costs. Regardless GA grantee the optimal alignment,

it requires more computation time with long DNA sequences greater than 200 residues.

In this paper, two techniques are proposed to deal with the pairwise DNA sequence alignment problem. The first is the Matching Regions process which it segments the two sequences into many windows then compare these windows with each other to produce matching and non-matching regions. While the second process is the Genetic Algorithm which its work based on the result of the first process. GA can be applied as an optimization tool search space that produces multiple solutions. In this paper, the large search space of non-matching regions breaks into several smaller subspace. Then assign each subset to GA in parallel. The concluding alignment score is the matching regions score plus the score of non-matched regions. As a result,the work has competence for computing the optimal alignment score.

The rest of the paper organized as the follows. Section2 consists of some of the related works. In section3 the theoretical part of the proposed work which discusses the Matching Regions Process and Multizone Genetic algorithm based with examples. Finally, Results and Conclusions are explained in Section 4 and Section 5, respectively.

2. Related works

Ranjani Rani, Dr. D. Ramyachitra[19]proposed a hybrid algorithm called Multi-Objective Bacterial Foraging Optimization Algorithm (MO-BFO) for solving Multiple sequence alignment problems. The proposed work consists of two algorithms, the first algorithm is Hybrid Genetic Algorithm with Artificial Bee Colony (GA-ABC), while the second is the Bacterial Foraging Optimization Algorithm. The proposed work employs four objective functions: maximize Similarity, Maximize the Conserved Blocks, minimizes the value of gap penalty, and Maximize Non-Gap Percentage to get the best sequence alignment.

Junsu Lee, YunkuYeu, Hongchan Roh, Youngmi Yoon, Sanghyun Park [20] proposed method based on the graph, in memory distributed system trinity for solving sequence alignment. the proposed method called Bulk Aligner which consist two stages. The first stage trims the reference sequence into s multiple segments of size s-1. Each segment converted into the graph by segment it into k-mer which form the node of the graph, then store the graph on memory cluster. The second step converts the query sequence into the graph and finds the longest paths to represent alignment result. This work takes more memory size.

Li, Ranka, and Sahni[21] developed a single-GPU parallelization based on the Smith-Waterman algorithm to solve the pairwise sequence alignment problem. the scoring matrix divided into strips then each strip works in parallel by using threads. the disadvantage of this algorithm is GPU-based and cannot be employed by CPU-based computers.

The primary drawbacks of the previous related works [19][20][21] and the others that introduced in the Introduction are they require more memory allocation and time-consuming. While the proposed work requires less memory size and CPU computing time due to reducing the alignment search space at matching Region process and due to the paralleling of the alignment space with the Genetic algorithm.

3. Proposed work

The proposed work consists of two techniques as mention in (section 1) the matching and MZGA techniques. The matching process segments the two sequences into multiple windows then compare these windows with each other to produce matching and non-matching regions (Section 3.1). The second step is assigning each non-matching region to the genetic algorithm in parallel. If the size of the non-matching region is big it breaks it in multiple regions, then assigns these regions to MZGA (Section 3.2). Fig. 1 show an overview of the proposed work that suggested in this paper.

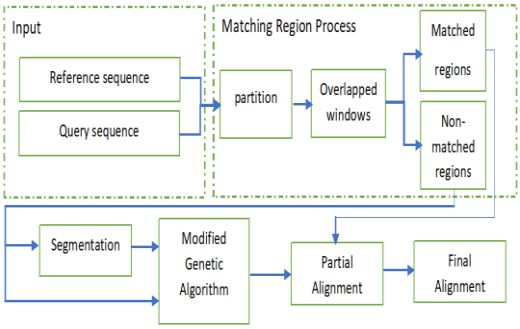

Fig. 1:the proposed system block diagram

3.1 Matching Regions Process

DNA is the hereditary material of organisms. DNA carries genetic information which stored as a code consisting of four chemical bases called nucleotide: adenine (A), cytosine (C), guanine (G), and thymine (T). Where nucleotide forms the chromosomes. Everyone has a unique sequence of DNA. Though there are no genetically identical individuals, there are exists a high similarity between them.

Assume there are two DNA sequences R and Q of unequal size. R is a reference sequence while Q is a query sequence. The proposed matching process is performed by two steps. The first step is the partition process which it segments the two sequences into multiple overlapped windows of equal size, while the second step is to look for matching and non-matching regions by comparing each window in Q with windows in R (see fig. 3 (a and b)) if matching is found then it labels it with match label otherwise it labels it with 0. The matching process explains in the following Algorithm.

Input: A pair of Sequences Q and R, Q is the query sequence of length x and R is the Reference sequence of y length.
Output: Matched and Non-matched region

1. Let:
2. Q_q and R_r form the working memory that corresponding to Q and R respectively.
3. s_q and s_r are the start location of each w_q and w_r windows respectively.
4. Match-labelis the label for each base in the matched region.
5. Set:
6. the initial value of a s_q and s_r to 0
7. the initial value of match-label to 1
8. All values of Qq and Rr to 0
9. A sequence Q and R are initially divided into overlapped w_q and w_r windows respectively, where w_s is the size of each window $w_{qi \text{ and }} w_{rj}$, $\forall i \in \{1, 2, \ldots, w_q\}$ and $\forall j \in \{1, 2, \ldots, w_r\}$ respectively
10. For each wq widow Do
11. For each wr window Do
12. If match(w_{qi}, w_{rj}) then
13. Label each base in Q_q and R_r that corresponding to w_{qi} and w_{rj} window with *match-label*
14. Increase the value of match_label by 1
15. Skip w_{s-1} of windows from w_q and w_r by set s_q and s_r
16. $sq \leftarrow sq + ws$
17. $sr \leftarrow sr + ws$
18. Stop the Internal loop.
19. Else
20. Skip current w_{rj} window and move s_r to the new window
21. $sr \leftarrow sr + 1$
22. End If
23. End For
24. If the current window w_{qi} *that compared with all windows in Reference and no matched window is found, then*
25. $sq \leftarrow sq + 1$

Fig. 2: Matching Regions Algorithm

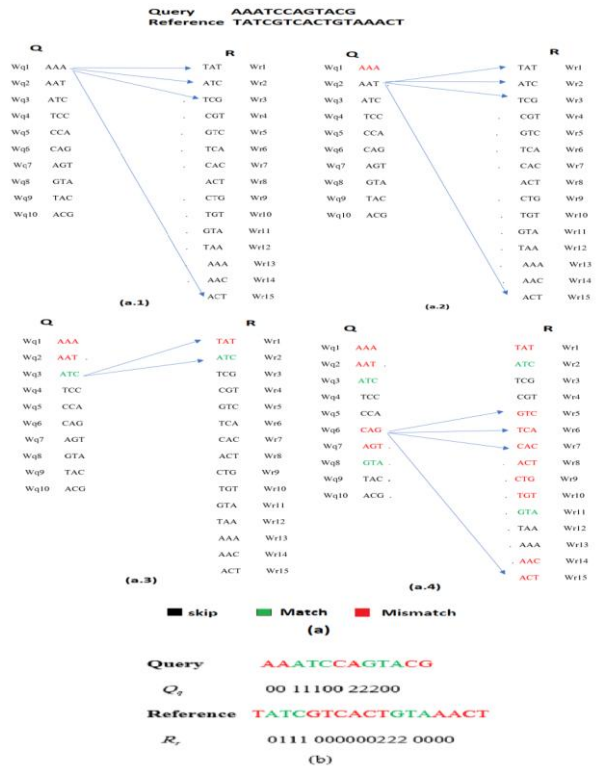

Fig. 3: A Sample of Matching Region Process. (a) Partitioning process where (a.1)First window wq1 of query sequence is scanned with all the reference windows and no matching window are found then it will be mismatched window;(a.2)The same process applied with wq2 window and no match is found; (a.3)wq3 window scanned with wr windows and match are found with wr2 then the two windows are labeled with match label and the next two windows of reference and query windows are skipped; (a.4)wq6 window will scan with reference windows from wr5 to the last window and no match are found, the same process continues until reach to the last wq window. (b) matching and non-matching region

3.2 Sequence Alignment with Multi-Zone Genetic Algorithm

Finding the maximum match score between two sequences is the primary goal of alignment. Gaps (referred to as "-") are used in an alignment. Gaps are embedded in both sequences to equal their length. A GA-based method was developed to solve the pairwise DNA sequence alignment problem. The affine gap penalty technique is used in the proposed work.

Naturally, extracted DNA Sequences length is variable. After matching Regions process, the proposed method considered sequences of non-matching regions are variable in size and in length. A partitioning method for alignment has been implemented on the large non-matching region. The purpose of partitioning is to reduce the complexity. The proposed method aligned sequence pair of each part on non-matched region in parallel and added it to the score of matching regions to get the final alignment score. Traditional Genetic Algorithm has been developed by changing the selection, and crossover processes.

3.2.1 Population

The population initialization is the first process in the Genetic Algorithm where the population forms by N individuals/ chromosomes. In the proposed work the N individuals are created in randomly way. Each individual/ chromosome in the population represents a possible alignment solution. Suppose each chromosome has size n then Each P*i*, $\forall i \in \{1, 2, \ldots, N\}$ is a series of an Integer value, for example, P*i=I1*, *I2*, …., *In*. Each seed*Ij*, $\forall j \in \{1, 2, \ldots, n\}$in P*i*, $\forall i \in \{1, 2, \ldots, N\}$ represent a gap position in the sequence that randomly chosen.

3.2.2 Sequence Encoding

In an alignment problem, gaps may be inserted in the sequences. Accordingly, the overall length of the alignment sequence is going to be the embedded gaps length plus the length of original sequence. In theproposed work theinserted gaps in one sequence is the same to the other sequence length. Suppose there are two sequences *R*and *Q*different in length. *R* is the longer sequence while *Q* is the shorter sequence or may be equal. ***Assume***, n1 and n2 are the lengths of R and Q respectively in the non-matching region, and ***nr*** and ***nq*** represent the number of gaps that inserted into *R, andQ* respectively. A single chromosome is formed based on *R* and *Q*. Thus, the chromosome length(*n) is* computed based on nr and nq as in equation 1.

$$n = (nr + nq), \quad (1)$$

$$nr = \| R \|$$
$$nq = \| Q \|$$

When the length of one or both two sequences of non-matching region n1 and n2 greater than 30 then the chromosome length will be increased. Thus, the computational complexity will increase. To overcome such a problem the proposed method divides the original sequences of non-matching regions into number of partitions (***p***).

The following is the sequence partition process:

1. A sequences R and Q of length $\mathbf{n_1}$ and $\mathbf{n_2}$, respectively are initially divided into *p* parts.
2. Assume $\mathbf{n_r}$ and $\mathbf{n_q}$ are the length of each part in R and Q, respectively. $\mathbf{n_{ri}} = (n1 / p)$, $\forall i \in \{1, 2, \ldots, \boldsymbol{p}_{-1}\}$ except for the last part $\mathbf{n_{rp}} = (n1 / p) + (n1\%p)$. Similarly,$\mathbf{n_{qi}} = (n2 / p)$, $\forall i \in \{1, 2, \ldots, \boldsymbol{p}_{-1}\}$ except for the last part $\mathbf{n_{qp}} = (n2 / p) + (n2\%p)$.

As an example of sequence partition and encoding, let smaller and unequal length sequences R and Q are considered here as in fig. 4.n1 equal to 47 while n2equal to 30. The method divides each sequence into parts, for example, three parts ($p = 3$). According to the sequence partition process, sequence R is divided into three parts R1,R2, and R3, and the length nr of each one is 15, 15,and 17, respectively. Similarly, sequence Q divided into three parts Q1,Q2 and Q3, and the length nq of each part is 10. (see Fig 4. 1(a)).

In an Integer Form, the sequences are encoded. Q_p1 and R_p1 are used as an example for the encoding process. The number of gaps nq that will be inserted to Q_p1 is 15 while the number of gaps nr that will have inserted to R_p1 is 10. As a result, the alignment length of Q_p1 and R_p1 is (25). While the chromosome length n is $(10 + 15)$(see fig. 4 (b)).

After decoding, the chromosome is represented by residues and gaps (indicated by "_") according to the integer values (see Fig. 4 (c)). The purpose of This conversion is to calculate the alignment score.The final alignment scores are the summation of all parts of the non-matching sequence and matching sequence.

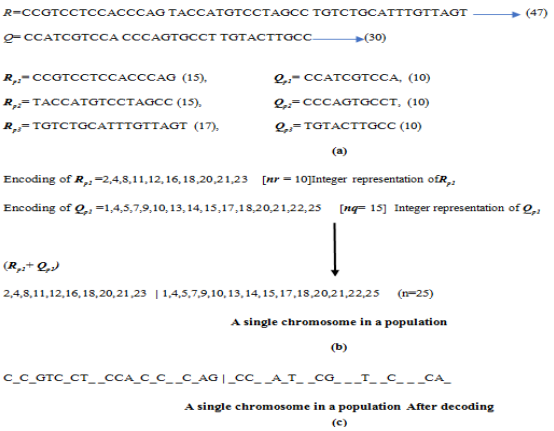

Fig. 4: Initial sequence representation in a population of smaller and short DNA segment (R and Q), and to illustrate the encoding process; (a) Segmentation Process; (b) Integer representation of a single chromosome in population; (c) single chromosome representation after decoding

3.2.3 Fitness function evaluation

The fitness function is one of the most important parts in Genetic Algorithm which is isolated from other parts that makes the genetic algorithm more robust and adaptive[22]. The fitness function is a criterion to measure the quality of the pairwise sequence alignment. The objective function to get the fitness score for each chromosome P*i*, $\forall i \in \{1, 2, \ldots, N\}$ in the population described in equation 2.

$$f = \sum_{i=1}^{t} w(Ri, Qi) \quad , \text{where } w = \begin{cases} +3, & \text{same nucleotide appears in R and Q} \\ -2, & \text{different nucleotide appears in R and Q} \\ -1, & \text{gap appears in R or Q} \\ 0, & \text{gap appears in R and Q} \end{cases} \quad (2)$$

And *t* is the total length of the chromosome after decoding.

3.2.4 Selection operations (Rank Based Multi Zone)

Selection is one of important genetic operation which selects two parents to produce offspring by crossover operation. As per Darwin's Theory of survival of the fittest, the best must survive and produce new offspring. In the proposed work the selection method divides the population into multi zones to ensure that the best individuals survive and produce new offspring. the proposed selection divides the population into three zones. The Mean of individuals fitness score is used as the criteria to divide the population into multi-zones. the process of select two parents P1 and P2 to produce two offspring C1 and C2 for the next generation is as the algorithm that shown in Fig. 5.

1. Rank the population descending based on their fitness score.
2. Compute mean1 and mean2 where,
   - 2.1. $\text{Mean1} = \frac{1}{N}(\sum_{i=1}^{N} fitness\ score\ i)$
   - 2.2 $\text{Mean2} = \frac{1}{N}(\sum_{i=x+1}^{N} fitness\ score\ i)$,x is the position of the last individual that his fitness score >= mean1.
3. Divide the population into three zones Z1, Z2, and Z3 based on mean1 and mean 2, where fitness score of individuals in Z1 is greater than or equal to mean, Z2 between mean1 and mean2, And Z3 smaller than mean2
4. Create the subset of individuals from Z1, Z2, and Z3. The Subset contains all individuals of Z1 and 30% and 20% that chosen randomly from Z2 and Z3, respectively.
5. While matching pool, not complete choose :
   - 5.1. P1 from Z1 randomly.
   - 5.2. P2 from the subset of Z1, Z2, or Z3 randomly.
6. End while

Fig. 5: Rank Based Multi ZoneAlgorithm

3.2.5 Crossover operator

Crossover is the fundamental operator in Genetic Algorithm that occurs after selection operation. The crossover strategy exchanges parents' information to produce new offspring. there are different strategies of the genetic algorithm[21]. In the proposed work the crossover operation takes two parents P1 and P2 from matching pool and exchange information in two regions between them to produce C1 and C2. The following algorithm shows the crossover process.

1. Select two regions r1 and r2 from R and Q, respectively from each parent to crossover, where the size of r1 and r2 is 50% of R and Q respectively.
2. Determine the start and end indexes of each region Randomly.
3. While not reach to the last seed in individual chromosome Do

3.1. If the seed in the range of r1 or r2 of P1 and P2 then
An exchange between their seeds to add new information to the child
3.2. Else
Copy each parent seed to the corresponding child
3.3. End if
End while

Fig. 6: Crossover Algorithm

3.2.6 Mutation Operator

Mutation operation is the last operation in MZGA which alters, inserts or deletes one or more seeds of an individual. In this paper, the proposed mutation operation takes one chromosome/individual Pi, $\forall i \in \{1, 2, \ldots, N\}$ randomly, then chose a gap position randomly from the chosen one. Finally check the next of the chosen gap if not a gap, shift the position of the gap by one seed. Therefore, the proposed method is represented algorithmically in Fig. 7.

Name: Matching Regions and Multi-Zone Genetic Algorithm
Input: Pair of DNA sequences R and Q which they are equal or different in length
Output: Optimal Alignment of Q and R Sequences.

1. Read a pair of DNA sequences R and Q.
2. Assign R and Q to the proposed Matching Regions process to get matched and Non-matched regions (as mention in section 3.1).
3. Evaluate the Similarity score for each matched region in R and Q.
4. Divide the Non-matched region of Q and R into smaller parts(section 3.2) and assign each part to the proposed Genetic Algorithm in parallel.
5. For each part *p* of the non-matched region on Q and R do
   - 5.1. Initial Randomly the population of size N. each individual P*i*, $\forall i \in \{1, 2, \ldots, N\}$ have size n, which is computed as described before in (section 3.2.2). the population length can be set by the user.
   - *5.2.* Evaluate the fitness score for each individual P*i*, $\forall i \in \{1, 2, \ldots, N\}$ of the population based on the objective function f*i (section 3.2.3).*
   - 5.3. Select two parents Randomly using depending on their fitness values for the population of N individuals.
   - 5.4. Perform crossover method as described before in section 3.2.5 between the chosen individuals P_1 and P_2.
   - 5.5. Each pair of parents P_1 and P_2 generates a pair of individuals called offspring (C_1 and C_2) to form a new pool of individuals as a population for the next generation.
   - 5.6. Mutant one individual of a population as described before in section 3.2.6.
   - 5.7. Terminate the process if reaches to the maximum number of iterations. Otherwise, go to step 5.2.
6. Compute the Similarity score for the non-matched part after the alignment process and then assign it to the total Similarity score.
7. Terminate the process if reached to the last part of non-matching parts.

Fig. 7: The proposed work algorithm

4. Results

Apply samples are taken from humans as a part or full of the gene. The DNA samples are extracted from cosmic (Catalogue of Somatic Mutations In Cancer) data set https://cancer.sanger.ac.uk/cosmic which consist 3328 normal genes and a huge amount of somatic mutation of cancer. Table1 summarizes the performance of the Matching Region process with different window size on 12 pairs of sequences of the Cosmic dataset. Table 2 summarizes the parameters setting that used by the proposed method. From (Fig. 8) it can be observed that the value of fitness tends to improve or converge in each successive generation. Finally, the work shows the ability for computing the optimal alignment score.

Table 1: Performance of Matching Regions process with different window sizes

No	(Normal_Gene(Reference_seq.) and_mutation_Gene(Query_seq.)) access ID	Reference_and_Query sequences length	Window size 20		Window size 40		Window size 60	
			Matched region	Non matched region	Matched region	Non matched region	Matched region	Non matched region
1	AF083811.1 COSM1257043	2157 2157	107	2	54	2	35	2
2	AF274025.1 COSM166763	654 654	31	2	16	2	11	2
3	AF282265.1 COSM6568930	2760 2746	136	2	67	2	44	2
4	AF282265.1 COSM4622218	2760 2760	137	2	68	1	45	1
5	AK026345.1 COSM5790617	801 801	40	1	19	2	12	2
6	AF395440.1 COSM5352525	195 194	9	1	4	2	2	1
7	AK131216.1 COSM1399519	1980 1978	98	2	49	2	33	2
8	ENST00000002596 COSM6305064	924 924	45	2	22	2	14	2
9	ENST00000002829 COSM3940575	2358 2358	117	2	58	2	38	2
10	ENST00000003084 COSM1684119	4443 4440	222	1	111	1	73	2
11	ENST00000003302 COSM5347280	3234 3219	160	2	79	2	53	2
12	ENST00000003583 COSM1745252	864 844	42	2	21	2	13	2

Table 2: Genetic algorithm parameter settings of the proposed method

Parameter	value
The population Size	30
Crossover	Two region crossovers
Selection type	Rank Based Multi Zone
Substitution Matrix	Match=+3; mismatch=-2
Gap penalty	Gap in one sequence =-1, in two sequence =0;
The Maximum Number of Generations	The Default is 1000 or set it by the user.

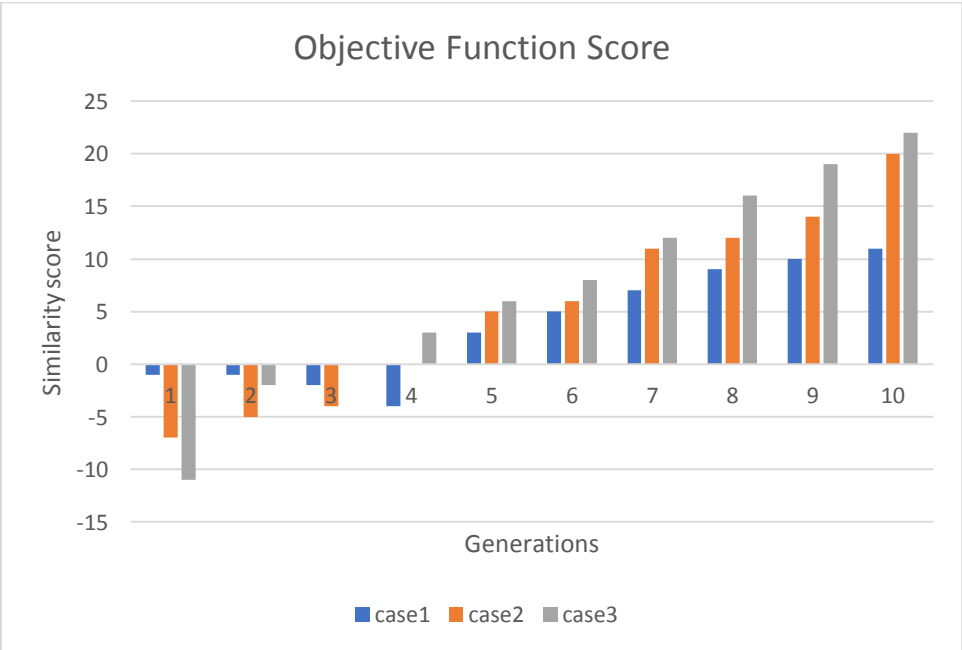

Fig. 8: Three cases of different DNA sequences length are implemented to show the variations in Fitness Score in different GA generations

5. Conclusion

Pairwise DNA sequence alignment is an important part of computation biology. In this work, the Matching Regions process is applied which determines only the non-matched region to go to the alignment process that will reduce the memory allocation. the segmentation process on the non-matched region reduce the searching space and have an advantage in the computation of the proposed method. The multi-Zone GA-based approach which was adopted to solve the problems experienced by the traditional GA in parallel rather than serial by using many threads. In addition to taking the problems that faced the DNA Sequence Alignment. The performance is enhanced due to the modifications of GA operation such as selection, crossover, and mutations schemes to align DNA sequences. The results show that the Zoning contributed to improving the fitness degree over the generations, rapid the process of convergence, and the exclusion of chromosomes can be expected to local minima or have bad solutions. Thus, obtaining a perfect gaps location and get the optimal pairwise sequence alignment. Fig 8. shows the variations in Fitness Score in different GA generations. Finally, it is worth mentioning that the proposed work proved to be generalizable. The work can also be extended to multiple sequences alignments.

References

[1] J. Xiong, Essential bioinformatics. Cambridge University Press, 2006.

[2] M. Gollery, "Bioinformatics: Sequence and Genome Analysis, David W. Mount. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press, 2004, 692 pp., $75.00, paperback. ISBN 0-87969-712-1.," Clin. Chem., vol. 51, no. 11, p. 2219, 2005.

[3] M. L. Metzker, "Sequencing technologies - the next generation.," Nat. Rev. Genet., vol. 11, no. 1, pp. 31–46, Jan. 2010.

[4] T. P. Quinn, I. Erb, M. F. Richardson, and T. M. Crowley, "Understanding sequencing data as compositions: an outlook and review," Bioinformatics, no. April, 2018.

[5] J. Sun, K. Chen, and Z. Hao, "Pairwise alignment for very long nucleic acid sequences," Biochem. Biophys. Res. Commun., vol. 502, no. 3, pp. 313–317, 2018.

[6] N. H. Kaghed, E. S. Al, F. Emad, and K. Al-Khuzaie, "Comparative study of Genetic Algorithm and Dynamic Programming of DNA Multiple Sequence Alignment," J. Babylon Univ. Appl. Sci. No, vol. 25, no. 2, 2017.

[7] P. Bonizzoni and G. Della Vedova, "The complexity of multiple sequence alignment with SP-score that is a metric," Theor. Comput. Sci., vol. 259, no. 1–2, pp. 63–79, 2001.

[8] W. Just, "Computational Complexity of Multiple Sequence Alignment with SP-Score," J. Comput. Biol., vol. 8, no. 6, pp. 615–623, Nov. 2001.

[9] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," J. Mol. Biol., vol. 215, no. 3, pp. 403–410, 1990.

[10] W. R. Pearson, "Rapid and sensitive sequence comparison with FASTP and FASTA," Methods Enzymol., vol. 183, pp. 63–98, Jan. 1990.

[11] D. W. Mount, "Using the Basic Local Alignment Search Tool (BLAST).," CSH Protoc., vol. 2007, p. pdb.top17, Jul. 2007.

[12] C. Kemena and C. Notredame, "Upcoming challenges for multiple sequence alignment methods in the high-throughput era," Bioinformatics, vol. 25, no. 19, pp. 2455–2465, Oct. 2009.

[13] B. Chowdhury and G. Garai, "A review on multiple sequence alignment from the perspective of genetic algorithm," Genomics, vol. 109, no. 5–6, pp. 419–431, Oct. 2017.

[14] M. A. Larkin et al., "Clustal W and Clustal X version 2.0," Bioinformatics, vol. 23, no. 21, pp. 2947–2948, Nov. 2007.

[15] T. L. Bailey and C. Elkan, "Fitting a Mixture Model by Expectation Maximization to Discover Motifs in Bipolymers," Proc. Second Int. Conf. Intell. Syst. Mol. Biol., pp. 28–36, 1994.

[16] P. Di Tommaso et al., "T-Coffee: a web server for the multiple sequence alignment of protein and RNA sequences using structural information and homology extension," Nucleic Acids Res., vol. 39, no. suppl_2, pp. W13–W17, Jul. 2011.

[17] I.-G. Mircea, I. Bocicor, and G. Czibula, "A Reinforcement Learning Based Approach to Multiple Sequence Alignment," in Soft Computing Applications, 2018, pp. 54–70.

[18] H. Nabeel Kaghed, S. Eman Al–Shamery, and F. E. K. Al-Khuzaie, "Multiple Sequence Alignment based on Developed Genetic Algorithm," Indian J. Sci. Technol., vol. 9, no. 2, 2016.

[19] R. R. Rani and D. Ramyachitra, "Multiple sequence alignment using multi-objective based bacterial foraging optimization algorithm," BioSystems, vol. 150, pp. 177–189, 2016.

[20] J. Lee, Y. Yeu, H. Roh, Y. Yoon, and S. Park, "BulkAligner: A novel sequence alignment algorithm based on graph theory and Trinity," Inf. Sci. (Ny)., vol. 303, pp. 120–133, 2015.

[21] J. Li, S. Ranka, and S. Sahni, "Pairwise sequence alignment for very long sequences on GPUs," 2012 IEEE 2nd Int. Conf. Comput. Adv. Bio Med. Sci. ICCABS 2012, 2012.

[22] S. K. Pal and P. P. Wang, Genetic algorithms for pattern recognition. CRC press, 2017.