IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 27, No. 1, February, 2022, pp. 67-74
Journal homepage: http://iieta.org/journals/isi

Semi Global Pairwise Sequence Alignment Using New Chromosome Structure Genetic Algorithm

Jeevana Jyothi Pujari[1*], Karteeka Pavan Kanadam[2]

[1] Computer Science and Engineering, VVIT, Acharya Nagarjuna University, Andhra Pradesh 522508, India
[2] RVR&JC College of Engineering, Chowdavaram 522019, Andhra Pradesh, India

Corresponding Author Email: jeevanajyothi.pujari@gmail.com

https://doi.org/10.18280/isi.270108

Received: 3 November 2021
Accepted: 12 January 2022

Keywords:
sequence alignment, genetic algorithm, optimal gaps, chromosome structure

ABSTRACT

Biological sequence alignment is a prominent and eminent task in the analysis of biological data. This paper proposes a pair wise semi global sequence alignment technique using New Chromosome Structure based Genetic algorithm (NCSGA) for aligning sequences by automatically detecting optimal number of gaps and their positions to explore the optimal score for DNA or protein sequences. The experimental results are conducted using simulated real datasets from NCBI. The proposed method can be tested on real data sets of nucleotide sequence pairs. The computational results show that NCSGA produces the near optimal solutions for semi global alignment compared to other existing approaches.

1. INTRODUCTION

The most prominent task in computational molecular biology is Sequence Alignment, in which two or more DNA/RNA/Protein sequence residues are aligned to identify the functional, structural and evolutionary relationships of biological sequences. Sequence alignment is a powerful tool in several fields such as illumination of functionally significant regions [1], phylogenetic reconstruction [2], RNA folding, prediction of structures of proteins and RNAs, homology between sequences, predictions of mutations [3], studying the properties of uncharacterized sequences and to suggest primers for PCR. When two sequences are arranged in a way that maximizes similarities or identities is called as pair wise alignment. An extension of pair wise sequence alignment is called as multiple sequence alignment in which more than two sequences are aligned [4]. Mainly sequence alignment solution depends on insertion of gaps [5]. There are three variants of pair wise sequence alignment which are Global, Local and Semi-Global sequence alignment. 1) If two sequences are aligned over their entire length termed as Global sequence alignment when it can be used where two sequences are more related and their sequence length is almost similar. 2) In local sequence alignment identifying highly conserved subsequences when the remaining part is divergent. 3) in some situations, one sequence is shorter than other, then it requires aligning short sequence against subsequence of bigger one, where trailing/leading gaps are ignored is called as Semi-Global sequence alignment or overlap alignment. Semi global sequence alignment algorithms are mostly used in aligning entire reads to a tiny fraction of genome in next generation sequencing. Applications involves sequence assembly, where it finds sequence fragments that overlap, that is we expect to be able to align the end of one fragment with the beginning of another, to find sequence fragments that start and end at the same position, used to find similarities that global alignments wouldn't and also used in aligning cDNA's with genomic DNA to identify gene structure [6]. However, semi global alignment has less work compared with the global and local alignments.

Dynamic Programming (DP) [7, 8] based approaches are most widely used approaches to solve pair wise alignment problems optimally such as Needleman-Wunch and Smith-Waterman approaches for global, semi global as well as local sequence alignment. Although, dynamic programming approach aligns global and semi global sequences optimally, but the complexity increases when many optimal paths are available. Computation cost also increases exponentially for aligning multiple sequences [9]. This makes the requirement of heuristic algorithms to align sequences optimally [10].

An adaptive heuristic based iterative search optimization technique, Genetic Algorithm (GA) is the popular Evolutionary approach, introduced by John Holland in 1975 [11]. GA was mainly proposed for simulating the biological evolution process, as embodiment of replication of human genetic process. The population of abstract solution is called as chromosome or genotype and each individual in a chromosome is called as gene. The collection of potential population is evolved in their multidimensional search space to find the best solution based on their objective function. The reasons for astounding the popularity of Genetic Algorithms are: its simplicity because it requires a primitive mathematical operator and its flexibility. GA's can be suitable for both numeric and combinatorial problems and other applications also [12]. Therefore, Genetic Algorithms can be effectively applied on various problems related to several fields. There have been several reviews on recent variants of GA's [13-15]. In past few years GA has been practiced in several exigent fields of bioinformatics [16].

Gap is a consecutive region occurs due to mutation of copying, insertion or replication of fragments in DNA sequence. Gaps and their positions play major role in assigning similarity scores to the alignment. The objective is to develop a new method for semi global sequence alignment by adding

or shuffling gaps automatically to maximize the similarity score. The performance of alignment algorithm is sensitive to number of gaps, choosing appropriate number of gaps and their positions are very critical or complex in practice. It is impossible to study the performance of the alignment algorithm for all possible number of gaps. This work proposes a new chromosome structure to determine optimal number of gaps automatically using Genetic Algorithm (GA).

This paper presents a Semi-global pair wise sequence alignment using Genetic algorithm with new chromosome structure. Proposed chromosome structure is employed in GA to solve semi global sequence alignment of sequences. The novelty of this algorithm is the variant of genetic algorithm by automatically choosing the number of gaps and their positions to explore more optimal solutions and accuracy and can be used as an optimization technique in a large complex search space to solve Semi Global sequence alignment problem efficiently. The simulation results show the significantly accurate results over the compared state-of-art and competitive algorithms.

The rest of the paper is classified as follows: Section 2 presents related work regarding sequence alignment Section 3 regarding preliminary concepts about semi global sequence alignment, scoring functions and genetic algorithm. Section 4 introduced the proposed method. Section 5 discuss about experimental Results and Conclusion.

2. RELATED WORK

Several techniques are presented for solving sequence alignment problem such as dynamic programming, progressive techniques, consistency-based approaches, and iterative algorithms. In Dynamic Programming (DP), for a pair wise sequence alignment it constructs two-dimensional matrix, where each dimension represents a sequence and the matrix can be filled according to scoring schema of matches, mismatches and gaps. The highest score is attained by performing backtracking on optimal diagonal path [17]. The DP-based Needleman-Wunch is used for global sequence alignment as well as for solving Semi-Global sequence alignment problem by modifying the algorithm as not penalizing starting/ending gaps [18], whereas Smith-Waterman algorithm is used for local sequence alignment [19]. Another DP-Matrix based approach to solve semi global sequence alignment problem is Gapmis [20] with a single gap and GapsMis [21] modifies the standard Needleman-Wunch algorithm to insert only specified bounded number of gaps in the alignment. The DP technique gives biologically optimal alignment score but practically requires more computation time when many optimal paths are available. As well as if we align the same sequences multiple times, static number of gaps and the gap positions are also static. The computational cost is also so high when more than three sequences are considered [22]. To reduce computational complexity and to explore more optimal solutions for this problem, several heuristic techniques are proposed. Heuristic techniques are categorized as progressive, consistency-based and iterative techniques. Progressive alignment is a popular technique to deal with the multiple sequences. In progressive based techniques [23-25], the distance matrix was constructed first to align closest sequence pairs and then the more distant ones are added. The guide tree constructed based on the distance matrix. The main drawback of this progressive method is once a sequence is misaligned then that error propagates to the end of the guide tree construction and will never be modified [26]. Other approach is consistency-based scoring scheme depends on the collection of libraries created to maintain the consistency score for both global and local pair wise alignments to produce consensus alignments. T-Coffee package [27] and DIALIGN [28] are the commonly used consistency-based approaches. This technique overcomes the greediness of the progressive approach but their computations are more complex and the usage of memory is also high.

To overcome the limitation of local optima in progressive approaches, various researchers proposed iterative based approaches which are applying evolutionary techniques heuristic in nature to solve sequence alignment problem. Iterative algorithms refine their search space through much number of times and give the equal priority for considering the sequences in the alignment. Othman et al. [29], Garai and Chowdhury [26, 30] proposed a Genetic algorithm-based approach for solving pair wise sequence alignment. Jangam and Chakraborti [31] developed a pair wise sequence alignment technique using the Ant Colony Optimization and the results are further aligned by GA.

However, all of the above studies try to align whole length sequences that is used to solve Global sequence alignment. Recently a few works have been concentrates on Semi-Global sequence alignment due to increase in demand for short read alignments in next generation sequencing. In Semi global sequence alignment finding optimal number of gaps and their positions for optimal alignment then it became a problem of optimization. such problems can be solved in better way using optimization algorithms. For instance, Barton et al. proposed a DP-Based approach for solving Semi-Global sequence alignment problem called as Gapmis [20], an algorithm for pair wise semi global sequence alignment with a single gap and GapsMis [21] which is suitable for aligning sequences with bounded number of gaps and it also produces optimal results for aligning short reads to reference sequence only. GPUGapsMis [32] is the parallel implementation of GapsMis. To the best of our knowledge this is the first effort of applying evolutionary based GA on Semi-Global Sequence alignment problem. This paper proposed a new chromosome structure-oriented population Genetic Algorithm to efficiently align sequences semi-Globally by automatically choosing appropriate number of gaps and the positions of gaps in the alignment.

3. PROBLEM DESCRIPTION: SEMI GLOBAL SEQUENCE ALIGNMENT

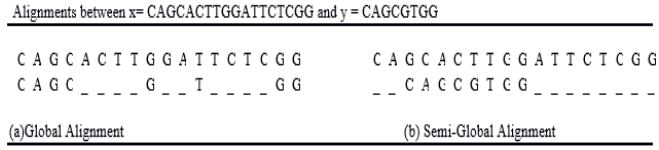

Figure 1. The alignments of Global and Semi-Global sequence alignment between the sequences x and y

Sequence alignment is a pervasive operation to map characters between DNA/RNA/Protein in a way of preserving order. It is mainly used to identify high similarity regions between sequences. The variant of global sequence alignment is used to find the potential overlap detection of entire short

sequence within longer sequence is called as Semi-Global pair wise sequence alignment, where it doesn't penalize the gaps at the beginning or ending of either of the sequences which is depicted in Figure 1.

For a given finite alphabet set $\sum$ consists of 4 nucleotides {A,T,G,C} for DNA and 20 amino acids for protein sequences and a Strings of *S1* and *S2* is a sequence of characters over $\sum$. *m* and *n* represent the length of sequences with $n \leq m$. For $1 \leq i \leq j \leq m$, the sub sequence of *S1* denoted by *ki ki+1… kj*. Matching similar regions by aligning one sequence to another sequence through insertion of special gap character as '_'. *S1'* and *S2'* can be obtained from *S1* and *S2* sequences through the insertion of gap symbols i.e. $S1, S2 \varepsilon \sum \cup \{-\}$. *l* is the length of the alignment, $min(S1,S2) \leq l \leq 1.20*min(S1,S2)$ in which beginning and ending gaps are not used to score in their objective function. Both of the aligned sequences may not contain gap characters with in the same i^{th} position. Computational approaches employing scoring functions to measure the similarity of sequences with the aim to maximize number of matches and minimize the number of gaps in the alignment. Score of two sequences *u* and *v* represented as *Z(u,v)* and the main objective is to find $max(Z(ki,j,S2)), where\ 1 \leq i \leq j \leq m$.

3.1 Genetic algorithm

GA is a metaheuristic, widely used optimization algorithm for solving hard and combinatorial problems in engineering and it is one of the most popular EA techniques, due to its good performance and easy to implement. A GA is a population based evolutionary algorithm where it produces randomly generated individuals called as chromosomes composed of genes and are represented as candidate solutions for the specified target problem. In general, better solutions evolve through several generations until a near optimal solution is reached. In the evolutionary process, every solution is estimated by fitness function of the specific target problem at each generation. The fitness function of target problem measures the goodness of each individual solution with respect to their objective requirements. During each generation some parent individuals (those usually having maximum fitness value) are selected based on their probabilistic approach, biased by fitness function. The genetic operators such as crossover and mutation operators are applied onto selected parent individuals to produce off springs, which inherit the features of parent individuals. The remaining solutions those having less fitness are discarded. On average, it is expected that, the fitness of the population will not decrease in every consecutive generation. This evolutionary process is repeated until a near optimal solution is obtained or some stopping criterion is reached such as the maximum number of generations and iterations.

4. PROPOSED METHOD: SEMI-GLOBAL SEQUENCE ALIGNMENT USING NEW CHROMOSOME STRUCTURE BASED- GENETIC ALGORITHM (NCSGA)

This paper proposed a new chromosome structure-based population genetic algorithm for semi global sequence alignment. The Genetic algorithm expresses the solution space as a collection of genotypes. In this paper, the genetic algorithm is designed to optimize the arrangement of residues and optimally choosing the number of gaps inserted into sequences. The structure of a new chromosome is encoded to optimize the number of gaps as one of the gene in the chromosome. The proposed algorithm is depicted in Figure 2 and the flow chart is described in Figure 3.

Algorithm: Semi Global Sequence Alignment using NCSGA

1. Read a pair of sequences (DNA/Protein) of length *m* and *n (n<m)*
2. Construct chromosomes of the following NCSGA structure.
3. Generate number of gaps G_n and randomly optimize the number of gaps between the scale factor 0 to 1.10
4. Randomly selects the possible alignment positions *k* ranges from 1 to *max(lgh(s1),lgh(s2))-min(lgh(s1),lgh(s2))+1 lgh(s1) lgh(s2)* are the length of sequence *s1* and *s2* respectively
5. Randomly generate initial population pop_j for all j belongs to {1…..N}
6. Evaluate the initial fitness of each chromosome in the population $\mathbf{pop_j}$ for all j {1.. N} using sum_of_pair score objective function.
7. Until max_gen reached
8. repeat
   a. A pair of individuals Px and Py in the population $\mathbf{pop_j}$ is selected based on their fitness score using Rank based selection method .
   b. Perform one-point cross over between selected individuals Px and Py and mutate the individuals to generate next generation population.
   c. Align the sequence shorter one on the longer one in the chromosome by using 'k', by randomly selecting the position of the large sequence from 1 to max(lgh(s1),lgh(s2))-min(lgh(s1),lgh(s2))+1 which is also encoded in the chromosome as one of the gene
   d. Evaluate the fitness of each individual by Similarity Score objective function.
9. record the best sum_of_pair score.

Figure 2. Proposed NCSGA algorithm

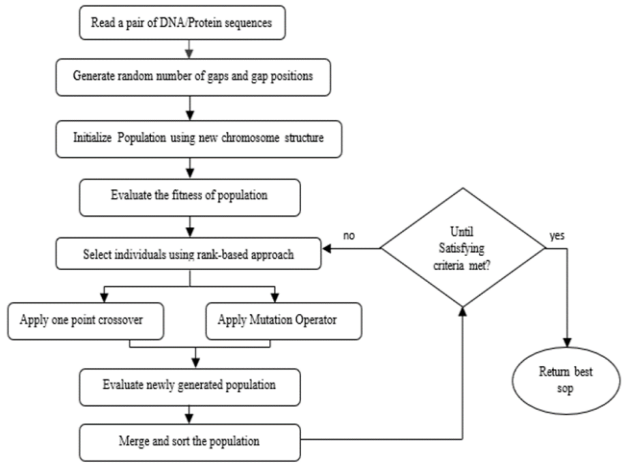

Figure 3. Flowchart of NCSGA algorithm

4.1 Semi global alignment with GA

The main objective of the proposed method is to automatically optimize the number of gaps to align short sequence as a pattern to large sequence in semi global sequence alignment. The sequence alignment can be done optimally in two ways either with or without inserting gaps in the alignment. Gaps may be inserted in the process of alignment dynamically and sometimes they are not present [26]. In the proposed method gaps are randomly selected between scale factor 1% to 1.10% [33]. The insertion of the gap may be at the end of the alignment or within the sequence depends on the chromosome structure. But the gaps at both ends are not penalized in semi-global sequence alignment. But introducing optimal number of gaps is an important problem at the time of aligning sequences. Applying Genetic algorithm new chromosome structure to store the number of gaps as one of its gene and the index position of substring as another gene effectively optimizes the number of gaps and also the possible alignment position in the larger sequence.

The performance of algorithm is sensitive to number of gaps choosing appropriate number of gaps is very critical or complex in practice. it is impossible to study the performance

of the algorithm for all possible number of gaps. This work proposes a new chromosome structure to determine optimal number of gaps automatically using Genetic Algorithm (GA).

4.2 Proposed chromosome structure

Chromosome structure representation is necessary to describe each candidate population in GA. The representation describes the problem in a structured way using GA and it also regulates the genetic operators. Chromosome is collection elements of real numbers within the specified domain of values. The real-coded GA representation is more efficient for utilization of CPU time than binary GA representation [34].

The structure of proposed new chromosome is shown in Figure 4. Consider *S1(x1,x2,…,xk….xm), S2(y1,y2,…….yn)* are two sequences of length m and n, where *m>0, n>0 and n<m*. *k* is the starting index of subsequence of length *n* in *S1* that is aligned with *S2*. The chromosome in GA is composed of four genotypes. The first gene stores *Gn* represents the number of gaps to be inserted in sequences are automatically and optimally chosen ranges from *'0' to '1.10*n'*. The second gene stores the value of *k* which is the possible alignment positions ranges from *1* to *m-n+1*. Third and fourth gene stores the index positions of gaps in sequence *S1* and *S2* which is equal to *2*Gn*. The length of new chromosome is *(2+2Gn)* that depends on number of gaps. Where $0 \leq k \leq (m-n+1)$ and $0 \leq g_n \leq 1.10*n$. *Gp1 & Gp2* are vectors of size *Gn* , which stores the positions of gaps in sequences *S1* and *S2* respectively.

G_n	k	p_1	p_2	…….	p_n	q_1	q_2	…..	q_n
Optimal number of gaps	Possible alignment position in sequence of length 'm'	'G_{p1}' Gap positions in sequence 'S_1'				'G_{p2}' Gap Positions in sequence 'S_2'			

Figure 4. New chromosome structure composed of four Genes

4.2.1 Initialization of population

In the initialization process, 'N' number of candidate solutions are generated randomly, but it is necessary to meet the restrictions in this population and that there is enough diversity between genes for the optimal convergence of GA. Each individual solution P_j is called as chromosome. The randomly initialized chromosomes can be seen in below Figure 5 and Figure 6. Initially each chromosome is formulated as *Gn* allowed number of gaps in both the sequences can be chosen dynamically in each iteration and *k*, which stores the starting index of subsequence of *S1* and ranges from *1* to *m-n+1*, The remaining genes in the chromosome represents the positions of gaps in both the sequences. The initialization of chromosome outlined below.

Step 1: for i=1 to n do
Step 2: for each chromosome, Set the initial gene as number of gaps 'Gn' and would be randomly chosen in the range of 0 to 1.10*n
Step 3: second gene 'k' as starting index of substring in sequence 'S1' ranges from 1 to 'm-n+1' can also be chosen randomly
Step 4: Third and fourth genes are randomly initialized vectors of storing the index positions of gaps in sequence 'S1' and 'S2'
Step 5: obtain the chromosome of having four genes with different sizes.

The above steps are followed to construct each chromosome and also repeated *n* times to generate initial population pop_j. The gap positions are chosen between length of short sequence that is *n*. For chromosome representation let us examine two sequences *S1* and *S2* of length 24 and 20. Initially the chromosome is formed as *k* which is randomly chosen as 3, gaps to be inserted are automatically chosen as 2, and the positions of gaps chosen within the length of short sequence. The total length of chromosome is (2+2* G_n) as (2+2*2)=6.

2	3	8	16	4	9
G_n	K	G_{p1}		G_{p2}	

Figure 5. One of the randomly initialized chromosome P_1

1	4	7	2
G_n	K	G_{p1}	G_{p2}

Figure 6. Another randomly initialized chromosome P_2

4.2.2 Objective function

The encoded genes in each chromosome can be applied to pair of sequences for scoring the alignment. Gaps are introduced in the sequences based on the elements stored in chromosomes. The alignment length after introducing gaps are $n+G_n$ in both of the sequences. After transformation of chromosome, the population can be represented by residues of two sequences and their gaps as a whole can be given as input to scoring function called as similarity score. Similarity Score, Column Score are the functions used to evaluate the fitness of semi global Sequence alignment.

Similarity Score.

The quality of pair wise sequence alignment can be assessed by similarity scores (SS), which determines the proportion of similarly aligned residue pairs in the alignment and can be formulated as in Eq. (1) and (2).

$$ss = \sum_{i=1}^{l} score(p_i, q_i) \tag{1}$$

$$score(p_i, q_i) = \begin{cases} \alpha & if\, p_i = q_i (match\ Score) \\ \beta & if\ (p_i \neq q_i) \wedge (p_i \vee \& q_i \neq {}' - {}') \\ \gamma & (mismatch\ Score) \\ & if\ (p_i \vee q_i = {}' - {}') \wedge (p_i \& \wedge q_i \neq {}' - {}') (gap\ cost) \end{cases} \tag{2}$$

where, *l* is the length of alignment, *p* and *q* are the two sequences used for test cases. Character position within the sequence is represented by *i*. *α, β* and *γ* are assigned to match score, mismatch score and gap score respectively. Numerical scores *α* and *β* are estimated by using substitution matrices such as NUC44, BLOSUM62 and PAM250 are the most popular matrices used for nucleotide, protein sequences respectively. where γ is the value assigned by the user as gap cost. Instead of using linear gap penalty, here we use to penalize the gaps using affine-gap penalty function in which initial gaps are penalized more than the extended gaps as in Eq. (3).

$$gap\ cost(\gamma) = gap_{open} + gap_{extend}(\gamma - 1) \tag{3}$$

Column Score.

It requires less complex computation to give the score for the fraction of identically aligned positions and can be

formulated as in Eq. (4) and Eq. (5).

$$CS = \sum_{i=1}^{l} m(p_i, q_i) \quad (4)$$

where,

$$m(p_i, q_i) = \begin{cases} 1 & if\ (p_i = q_i) \\ 0 & otherwise \end{cases} \quad (5)$$

4.2.3 Genetic operators

As in Genetic Algorithm three operators such as selection, crossover and mutation operators are used.

Selection operator.

After calculating the fitness score of each chromosome in the population, selecting chromosomes as parents for reproduction in the current generation is very important because it improves the quality of populations in successive generations. There are two types of selection schemes known as proportionate scheme and ordinal based scheme [35]. Proportionate scheme selects the chromosomes based on their fitness values relative to the fitness of other chromosomes in the population whereas ordinal based scheme selects chromosomes based on the ranks given within the population. We implemented one of the proportionate selection schemes in each selection is Roulette Wheel selection where the chromosomes having best alignment similarity scores have been selected as parents for reproduction.

Cross-over.

Cross over is an essential operator for generating new off springs in the next generations. Many techniques are there to perform the crossover operation such as single point, multi point, gene-by-gene crossover [36]. We selected the gene-by-gene crossover operator and adapted to the application due to its specific constraints imposed on the chromosome structure. The steps involved in the crossover operation are shown as follows.

Step 1. Randomly selects two individuals as $parent_a$ and $parent_b$ to produce offspring solutions. Each chromosome having equal number of genes in their encoding structure.

Step 2. With the cross over probability of 0.5 decides the fragments of the genes are exchanged or stays as resides.

a. If $len(parent_a = parent_b)$ then the fragments of genes are directly exchanged in both the parents.

b. If $len(parent_a \neq parent_b)$ then a contraction or expansion process must be followed to generate children to meet the complex constraints specified in the problem and to produce the feasible solutions. In expansion process, the number of gaps in each sequence can be padded randomly and must be matched with the second gene of the chromosome. In contraction process, the fragments are gene are removed when excess number of gaps can be found.

Mutation.

To explore the search space, mutation operator is implemented on chromosomes [37]. By changing the gap position in the alignment of the chromosome for yielding a better off spring with the mutation rate as 0.001. During the mutation the number of gaps can be altered in the chromosome but the total number of residues in the population should be equal to the length of shorter sequence. The positions of the gaps in the selected chromosome are chosen randomly based on the mutation rate and perform alteration of gap positions gene by gene. This mutation process is applied for the individuals until the elite population is reached.

5. EXPERIMENTAL RESULTS

The main purpose of this algorithm is to optimally finding number of gaps in the alignment in order to maximize the Similarity score. To highlight the suitability of this technique, the results are compared with the Emboss-needle which implements Needleman wunch algorithm to align sequences semi globally, Gap-mis is a tool to align sequences with a single gap and another tool Gaps-mis which allows bounded number of gaps in the alignment iteratively.

The following parameters chosen to perform this experiment. The population size used in NCSGA pop_j=200 and the number of generations max_{Gen}=50 is used to get the optimal similarity score for semi global sequence alignment. In the Similarity score nuc44 is the substitution matrix is used and for scoring the gaps, Gap_{open}=10 and Gap_{extend}=0.5.

In order to evaluate the performance of the proposed algorithm, we collected the datasets from NCBI database [38]. From the database, Arabidopsis Thaliana (AT) Chromosome1 and Burkholderia Glumae (BG) genomic data were selected. We randomly select sequences from the specified genomes and further process the sequences by introducing the bases into the reference sequences to produce synthetic data, yet it is taken from the real data. It is more realistic than the randomly generated data. The performance analysis on the randomly generated data from real sequences is good because the algorithm treats the data as equal [21].

For our experiments, we simulated real sequences that is produced from the above specified genomic data as {50,100,200,250} bp as reference sequences. By inserting mismatches and gaps into these sequences produces query sequences of more length. The query sequences can be aligned back to reference sequences. From those we measure the performance of semi globally aligned sequence pairs in the set.

The similarity scores of different algorithms can be formulated as in Table 1. We collected 20 pairs of sequences of each query length sequences. That is, we perform experiments on 20 pairs of each 50,100,200,250,300 and 400bp length sequences.

From Table 1, we observed that if we simulate the real sequences by inserting mismatches and gaps in the sequences then NCSGA performs well by inserting appropriate gaps in the alignment automatically. Gapmis and Gapsmis allows a static number of gaps in the alignment. An Emboss- needle algorithm produces a single alignment solution and the score. The best score was obtained by NCSGA for all the sequence pairs compared to Gapmis. NCSGA performs equally well compared to Emboss-needle and Gapsmis and also produces many optimal alignments solutions due to the gap dynamic gap positions in the sequences.

We tabulated the results of synthetic data of different pairs of different lengths in Table 2. From Table 2, we observed that, NCSGA performed better results in these 10 sequence pairs compared to other specified algorithms. It is notable that, if the lengths of the sequences are different and those are much divergent to each other, NCSGA gives more optimal alignments and best semi global alignment similarity scores.

Table 1. Average similarity scores of semi globally aligned DNA sequences using NCSGA, Emboss-Needle, Gapmis, Gapsmis Algorithms

Species	Length of Query sequence[bp]	Emboss-needle	Gapmis	Gapsmis	NCSGA
AT	50	250	250	240	250
	100	491	491	481	491
	200	991	991	981	991
	250	1250	1250	1240	1250
BG	50	250	250	240	250
	100	500	500	490	500
	200	991	991	981	991
	250	1250	1250	1240	1250
	300	1500	1490	1481	1490
	400	1500	1500	1490	1500

Table 2. Similarity scores of DNA sequences with NCSGA and compared algorithms

S.No.	Seq no.	Gapmis	Gapsmis	GA	NCSGA
1	pair1	-154	-126	-15	**2**
2	pair2	-13	-21	8	**18**
3	pair3	-4	-5	-10	**5**
4	pair4	-16	-23	11	**19**
5	pair5	-45	-36	2	**5**
6	pair6	-21	-12	4	**15**
7	pair7	-7	-7	6	**14**
8	pair8	-58	-22	-44	-35
9	pair9	-12	-10	5	**18**
10	pair10	-30	-22	7	**15**

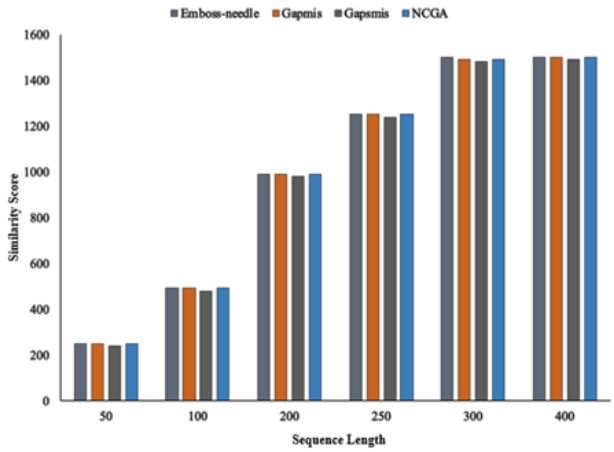

Figure 7. Comparative results of similarity scores for emboss needle, Gapmis, Gapsmis, NCSGA techniques

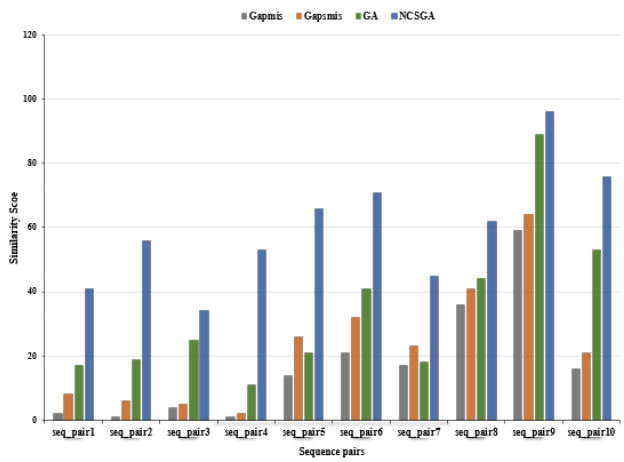

Figure 8. Comparative results of average similarity scores for Gapmis, Gapsmis, GA, NCSGA methods comparative

Figure 7 depicts the performance comparison of emboss-needle, Gapmis, Gapsmis and NCSGA. From the Figure 7 it is further noted that the NCSGA Algorithm Performs equally with the other algorithms by exploring more optimal solutions with better accuracy. Figure 8 also depicts the performance of NCSGA over other comparative algorithms in terms of similarity scores. In Figure 8 we noticed that, If the sequences are more divergent to each other, then also the proposed NCSGA yielded optimally best similarity scores for the sequence pairs.

6. CONCLUSIONS

In this paper, we focused on aligning sequences semi globally by a new chromosome structure based Genetic algorithm (NCSGA). Evolutionary algorithms are proved to be most successful algorithms in many domains including sequence alignment. Number of gaps is a priori in all those methods which impacts algorithm accuracy. But in practice it is very difficult to automatically predict the number of gaps in the alignment. The novel structure of chromosome encoding in GA automatically optimizes the number of gaps and their positions of gaps in the alignment, as well as the position of longer sequence to be align with the shorter one. Experimental results have shown that NCSGA achieved best accuracy in divergent sequences as compared to existing algorithms for semi global sequence alignment. The included parameters in the new chromosome structure enhances more search space for leading optimal semi global alignments. Though it is proved as the best among the said algorithms, it requires more computational time for lengthy sequences instead accuracy is given a high priority. The work can be further enhanced for multiple sequences.

REFERENCES

[1] Pei, J. (2008). Multiple protein sequence alignment. Current Opinion in Structural Biology, 18(3): 382-386. https://doi.org/10.1016/j.sbi.2008.03.007

[2] Wang, L.S., Leebens-Mack, J., Wall, P.K., Beckmann, K., DePamphilis, C.W., Warnow, T. (2009). The impact of multiple protein sequence alignment on phylogenetic estimation. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 8(4): 1108-1119. https://doi.org/10.1109/TCBB.2009.68

[3] Hicks, S., Wheeler, D.A., Plon, S.E., Kimmel, M. (2011). Prediction of missense mutation functionality depends on both the algorithm and sequence alignment employed.

Human Mutation, 32(6): 661-668. htttps://doi.org/10.1002/humu.21490
[4] Xiong, J. (2006). Essential Bioinformatics. Cambridge University Press.
[5] Katoh, K., Standley, D.M. (2016). A simple method to control over-alignment in the MAFFT multiple sequence alignment program. Bioinformatics, 32(13): 1933-1942. https://doi.org/10.1093/bioinformatics/btw108
[6] Durand, D. (2015). Database Searching and BLAST Tuesday, October 27th.
[7] Needleman, S.B., Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48(3): 443-453. https://doi.org/10.1016/0022-2836(70)90057-4.
[8] Smith, T.F., Waterman, M.S. (1981). Identification of common molecular subsequences. Journal of Molecular Biology, 147(1): 195-197. https://doi.org/10.1016/0022-2836(81)90087-5
[9] Azim, G.A., Hamdi-Cherif, A. (2008). Pairwise sequence alignment revisited-genetic algorithms and cosine functions. Applied Mathematics, Simulation, Modelling, 2008.
[10] Bucak, I.Ö., Uslan, V. (2010). An analysis of sequence alignment: Heuristic algorithms. In 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, pp. 1824-1827. http://dx.doi.org/10.1109/IEMBS.2010.5626428
[11] Holland, J.H. (1992). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. MIT Press. http://dx.doi.org/10.7551/mitpress/1090.001.0001
[12] Goldberg, D.E. (1989). Genetic Algorithms in Search. Optimization, and Machine Learning. Addison-Wesley Longman Publishing Co., Inc.
[13] Abuiziah, I., Nidal, S. (2013). A review of genetic algorithm optimization: Operations and applications to water pipeline systems. International Journal of Physical, Natural Science and Engineering, 7(12): 341-348.
[14] Patil, V.P., Pawar, D.D. (2015). The optimal crossover or mutation rates in genetic algorithm: A review. International Journal of Applied Engineering and Technology, 5(3): 38-41.
[15] Elsayed, S.M., Sarker, R.A., Essam, D.L. (2010). A comparative study of different variants of genetic algorithms for constrained optimization. In Asia-Pacific Conference on Simulated Evolution and Learning, pp. 177-186. https://doi.org/10.1007/978-3-642-17298-4_18
[16] Manning, T., Sleator, R.D., Walsh, P. (2013). Naturally selecting solutions: the use of genetic algorithms in bioinformatics. Bioengineered, 4(5): 266-278. https://doi.org/10.4161/bioe.23041
[17] Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G. (1998). Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press. http://dx.doi.org/10.1017/CBO9780511790492
[18] Chen, J., Guo, M., Wang, X., Liu, B. (2018). A comprehensive review and comparison of different computational methods for protein remote homology detection. Briefings in Bioinformatics, 19(2): 231-244. https://doi.org/10.1093/bib/bbw108
[19] Smith, T.F., Waterman, M.S. (1981). Identification of common molecular subsequences. Journal of Molecular Biology, 147(1): 195-197. http://dx.doi.org/10.1016/0022-2836(81)90087-5
[20] Flouri, T., Frousios, K., Iliopoulos, C., Park, K., Pissis, S., Tischler, G. (2013). GapMis: A tool for pairwise sequence alignment with a single gap. Recent Patents on DNA & Gene Sequences, 7(2): 84-95. http://dx.doi.org/10.2174/1872215611307020002
[21] Barton, C., Flouri, T., Iliopoulos, C.S., Pissis, S.P. (2015). Global and local sequence alignment with a bounded number of gaps. Theoretical Computer Science, 582: 1-16. https://doi.org/10.1016/j.tcs.2015.03.016
[22] Notredame, C. (2002). Recent progress in multiple sequence alignment: A survey. Pharmacogenomics, 3(1): 131-144. https://doi.org/10.1517/14622416.3.1.131
[23] Hogeweg, P., Hesper, B. (1984). The alignment of sets of sequences and the construction of phyletic trees: An integrated method. Journal of Molecular Evolution, 20(2): 175-186. http://dx.doi.org/10.1007/BF02257378
[24] Thompson, J.D., Higgins, D.G., Gibson, T.J. (1994). CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic acids Research, 22(22): 4673-4680. https://doi.org/10.1093/nar/22.22.4673
[25] Larkin, M.A., Blackshields, G., Brown, N.P., Chenna, R., McGettigan, P.A., McWilliam, H., Higgins, D.G. (2007). Clustal W and Clustal X version 2.0. Bioinformatics, 23(21): 2947-2948. https://doi.org/10.1093/bioinformatics/btm404
[26] Garai, G., Chowdhury, B. (2015). A cascaded pairwise biomolecular sequence alignment technique using evolutionary algorithm. Information Sciences, 297: 118-139. https://doi.org/10.1016/j.ins.2014.11.009
[27] Notredame, C., Higgins, D.G., Heringa, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. Journal of Molecular Biology, 302(1): 205-217. https://doi.org/10.1006/jmbi.2000.4042
[28] Morgenstern, B. (1999). DIALIGN 2: Improvement of the segment-to-segment approach to multiple sequence alignment. Bioinformatics (Oxford, England), 15(3): 211-218. https://doi.org/10.1093/bioinformatics/15.3.211
[29] Othman, M.B., Hamdi-Cherif, A., Azim, G.A. (2008). Genetic algorithms and scalar product for pairwise sequence alignment. International Journal of Computers, 2(2): 134-147.
[30] Garai, G., Chowdhury, B. (2012). A novel genetic approach for optimized biological sequence alignment. Journal of Biophysical Chemistry, 3(2): 201-205. https://doi.org/10.4236/jbpc.2012.32022
[31] Jangam, S.R., Chakraborti, N. (2007). A novel method for alignment of two nucleic acid sequences using ant colony optimization and genetic algorithms. Applied Soft Computing, 7(3): 1121-1130. https://doi.org/10.1016/j.asoc.2006.11.004
[32] Carroll, T.C., Ojiaku, J.T., Wong, P.W. (2019). Semiglobal sequence alignment with gaps using GPU. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 17(6): 2086-2097. http://dx.doi.org/10.1109/TCBB.2019.2914105
[33] Chellapilla, K., Fogel, G.B. (1999). Multiple sequence alignment using evolutionary programming. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 1: 445-452.

http://dx.doi.org/10.1109/CEC.1999.781958
[34] Herrera, F., Lozano, M., Verdegay, J.L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artificial Intelligence Review, 12(4): 265-319. https://doi.org/10.1023/A:1006504901164
[35] Hüe, X. (1997). Genetic algorithms for optimization. Technical report, The University of Edinburgh.
[36] Umbarkar, A.J., Sheth, P.D. (2015). Crossover operators in genetic algorithms: A review. ICTACT Journal on Soft Computing, 6(1): 1083-1092. https://doi.org/10.21917/ijsc.2015.0150
[37] Pawar, S.N., Bichkar, R.S. (2015). Genetic algorithm with variable length chromosomes for network intrusion detection. International Journal of Automation and Computing, 12(3): 337-342. https://doi.org/10.1007/s11633-014-0870-x
[38] http://www.ncbi.nlm.nih.gov/Genbank/GenbankSearch.html. One of the main sequence databases.