DNA SEQUENCE RECONSTRUCTION BASED ON GENETIC ALGORITHM

Md. Rafiqul Islam, Md. Rowshan Shahriar, and Abul Faisal Mohammad Shaheed
Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh
Email: {dmri1978, shahriarmurad, shaheed02174}@yahoo.com

ABSTRACT

It is becoming increasingly important to develop a novel process for determining the letters of our genetic code, known as DNA sequencing. This task is performed by large datasets using the combination of heuristic methods with little mathematical calculation. In this paper, we present a new method for DNA sequence reconstruction using genetic algorithm, which is able to predict the actual DNA sequence. The performance of the genetic algorithm is evaluated with respect to previous methods in the literature. The results indicate that the proposed new method is superior to previous methods. Finally we compare the results of the experiment and discuss the performance of the proposed method in the DNA sequence reconstruction.

Keywords: DNA Sequence, Subsequence, Oligonucleotide, Population Construction, Fitness Function, Crossover, Mutation.

1.0 INTRODUCTION

DNA molecule, the fundamental fabric of life carries the genetic instruction for making living organisms. The main role of DNA is the long-term storage of information and it is often compared to a set of blueprints, since DNA contains the instructions needed to construct other components of cells, such as proteins and RNA molecules. The DNA segments that carry this genetic information are called genes, but other DNA sequences have structural purposes, or are involved in regulating the use of this genetic information. Chemically, DNA is a long polymer of simple units called nucleotides, which are held together by a backbone made of alternating sugars and phosphate groups. Attached to each sugar is one of four types of molecules called bases. The bases are adenine (A), thymine (T), cytosine (C) and guanine (G). It is the sequence of these four bases along the backbone that encodes information. This information is read using the genetic code, which specifies the sequence of the amino acids within proteins. Within cells, DNA is organized into structures called chromosomes and the set of chromosomes within a cell make up a genome. These chromosomes are duplicated before the cells divide, in a process called DNA replication.

In chemical laboratory methods of DNA sequencing, a long DNA strand is chopped into slices and then biochemical method is applied to detect the nucleotide sequence in a single fragment. Then these known fragments are used to form a superstring, which is almost the same as the original DNA strand. In 1994, Leonard Adleman used molecular tools to solve a hard computational problem [3]. Adleman used traveling salesperson problem to manipulate DNA sequence and this is the first solution of a mathematical problem with biological tools. In [1] J. Blazewicz *et al.* presented a method of DNA sequencing. In their method, if the hybridization experiment were executed without errors, the subsequence would contain nothing but all subsequences of length l of the original sequence of the known length n. Then the subsequence consists of $n-l+1$ elements and neighboring elements always overlap on $l-1$ nucleotides. In [3] S-M Chen *et al.* presented a method of DNA sequence with divide and conquer technique where they used a standard scheme of selection operation called the roulette wheel method.

The limitations of the conventional DNA sequencing methods can be overcome in the hybridization technique with the help of genetic algorithm; where in a single biochemical experiment (using a DNA probe chip) is possible to know almost all the subsequences of a specified length, which are presented in the original sequence. S-M. Chen *et al.* presented the method to solve DNA sequence alignment using genetic algorithm [3]. Inspired from their paper we present a new method for DNA sequence reconstruction using genetic algorithm. We choose the right successor with the help of successor choice technique and the fitness function. This method also deals with the oligonucleotides with negative and positive errors. Through this method we also manage the duplication of oligonucleotides.

2.0 EXISTING METHODS FOR DNA SEQUENCING

Reconstruction of the original DNA sequence due to a large number of possible combinations requires a computational support. In this paper, a new method of sequencing has been proposed. The method has been implemented and tested for the case of an ideal experiment. Though a number of researches have been conducted regarding DNA sequencing, but in the specific field of reconstruction the number of available researches is not enough. The following sections describe the most recent research developments on this aspect.

DNA sequencing; Tabu and Scatter search combined method was proposed by J. Blazewicz *et al.* in 2004 [1]. In their method, inserted or shifted oligonucleotides are stored by sorting them on the Tabu list for a given number of iterations. The list is checked for an attempt to shift or delete an oligonucleotide and prevented if the oligonucleotide is on the list. It does not appear necessary to remember clusters shifted in order to avoid cycling, since clusters often change after a move. An element found on the Tabu list may be deleted or shifted together with the cluster containing it and may also be deleted if there is no feasible move. In this case, an element presented in the Tabu list for the greatest number of iteration is chosen. There is a global criterion function to maximize the number of spectrum elements making the solution. A function called condensation that is able to compare all kinds of move for each solution comparing the number of oligonucleotides from the spectrum in the solutions. If the moves were compared by the global criterion function, deletions or shifts would be used very rarely. This method cannot provide high quality results. It also performs poorly for a long input DNA sequence. Moreover the proposed method did not use any additional information about spectral or original sequences, which could be derived from biochemical experiments.

A heuristic managing errors for DNA sequencing was proposed by J. Blazewicz *et al.* in 2002 [6]. They presented a new approximation algorithm for reducing both positive and negative errors. In their method, the presence of negative errors in a spectrum forces the overlapping between some neighboring oligonucleotides in a sequence on less than $l\text{-}1$ letters. Positive errors in a spectrum force the rejection of some oligonucleotides during reconstruction process. Benefits of this method are, it permits decreasing both negative and positive errors and brings a way of solution strategy for managing repetitions. This method works on a fitness function, which depends on the thickness of the superstring. For two different oligonucleotides of M and $M^{/}$ of length l, M is predecessor of $M^{/}$, if k $(1 \le k \le l - 1)$ the last nucleotides of M are the same as the first k nucleotides of $M^{/}$. The thickness of an oligonucleotide is the mean value of the all its nucleotides. Large thickness value is better to its current position for fitting into the reconstructed pattern. This method provides a reliability factor for each production. This method performs poor for a long DNA sequence and proper successor choice.

Sequencing by hybridization: an enhanced crossover operator for a hybrid genetic algorithm was proposed by Carlos A. Brizuela *et al.* in 2007 [9]. In their method, a genetic algorithm to deal with the computational part of the SBH problem is introduced. This algorithm is a variant of the hybrid genetic algorithm (HGA) proposed by J. Blazewicz *et al.* [11]. The variant achieves better similarity results, with respect to the original sequences for computationally hard instances, and shorter computation time. The input for the computational part of the SBH problem consists of a set $S = \{S_{1,} S_{2,} \ldots\ldots\ldots\ldots, S_k\}$ of equal length (l) strings S_i over the alphabet $\sum = \{A, C, G, T\}$, and a number n representing the length of the unknown sequence. Each S_i is always a fragment of the original sequence N, whenever the experiment is error free ($|S| = n - l + 1$). Here, S_i may represent a fragment that is not in the original sequence (positive errors). There may be fragments in the original sequence N that do not appear as a string S_i in S (negative errors). The problem is to find a sequence L of length no greater than n such that the number of used strings S_i is maximized, and therefore the difference between N and L is minimized. Each individual i is represented by a permutation of indices of oligonucleotides in the spectrum. The feasible solutions are represented by sub cycle free permutations, except for a single cycle of length $|S|$. For a feasible individual i = [4 2 3 0 1] and a given spectrum S = {CTG, ACT, GGA, GAC, TGA}. The number 4 at locus 0 indicates that the oligonucleotide at position 0 (CTG) in the spectrum, is followed by the oligonucleotide at position 4 (TGA).

In this way, following the indices in the spectrum, the sequence of oligonucleotides that individual i represents is: CTG, TGA, ACT, GGA, GAC (0, 4, 1, 2, 3, 0). For object function computation, first start at position zero of the resulting cycle (0, 4, 1, 2, 3, 0). Then joined the oligonucleotides, once the sequence has 3 oligonucleotides (CTG, TGA and ACT) and $n^{/} \le 6 \le n$, the next oligonucleotide (GGA) can be added. In this case, the resulting sequence with 4 oligonucleotides is CTGACTGGA (0, 4, 1, 2). The last oligonucleotide (GGA) has to be eliminated since it makes the sequence to be of length ($n^{/}$) greater than n. Hence, the number of oligonucleotides used when starting at position 0 is 3. The process is repeated starting at each locus in the chromosome. For each individual its fitness value is normalized, based on the maximum number of oligonucleotides in any valid sequence, and then linearly scaled, $f_{new} = \left(\frac{f}{n - l + 1} \right) k$ where, f is the number of used oligonucleotides, and k the scaling factor. For crossover, the best successor in the parents is always selected as long as a sub cycle is not generated. Otherwise, the best successor not generating a sub cycle is selected among the remaining oligonucleotides in the spectrum.

3.0 DNA SEQUENCE RECONSTRUCTION WITH SUCCESSOR CHOICE

3.1 Basic Concepts

In this section, a new method for DNA sequence reconstruction based on genetic algorithm is presented. It uses genetic algorithm for reconstructing the main sequence from its subsequences. It can reduce the time complexity to deal with the sequence reconstruction. A DNA probe chip is a biochemical chip, which contains all possible combinations of subsequences or probes (of a specified length *N*) of nucleotide bases. Hence, the total number of subsequences in a DNA probe chip will be 4^N. For example, if *N* equals *4*, then the probe chip will contain all possible subsequences of length *4*, starting from AAAA to CCCC. The subsequences on the chips are built by combinatorial chemical synthesis [4]. If *S* is a sequence, then we will have the set of subsequence of length *4*. The subsequence can be represented as shown in [3] as follows:

$S = \{Si, i = 0, 1, 2, \ldots\ldots\ldots\ldots, N\text{-}1\}$, where, S_i denotes the DNA subsequences present in the set.

We also include successor choice procedure to produce correct sequence. The main idea behind this proposed method is to implement DNA sequence reconstruction from a given sequence with appropriate successor choice.

The Reconstruction procedure will be performed using the following steps:

Step 1: Generating DNA subsequences.
Step 2: Application of the stages of the genetic algorithm.
Step 3: Generating DNA sequences with successor choice.

The flow chart of DNA sequence reconstruction process has been depicted in the following Figure 1.

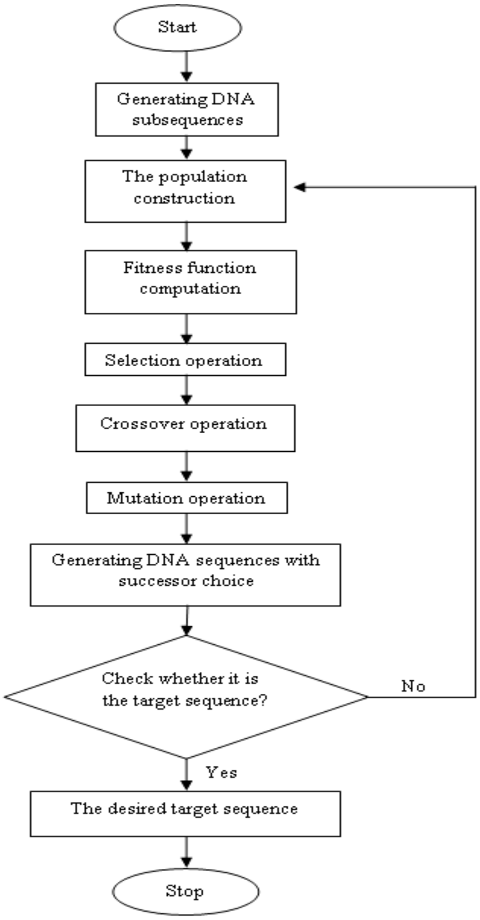
Start
Generating DNA subsequences
The population construction
Fitness function computation
Selection operation
Crossover operation
Mutation operation
Generating DNA sequences with successor choice
Check whether it is the target sequence?
No
Yes
The desired target sequence
Stop

Fig. 1: Flow chart of DNA sequence reconstruction

3.2 Generating DNA Subsequences

From a given DNA sequence, we first create all the possible combinations of subsequences of nucleotide bases. If the sequence is GACCATCGGA, then the subsequences with four bases will be GACC, ACCA, CCAT, ATCG, TCGG, and CGGA. Then we randomly select a base from the main sequence to produce subsequences. In hybridization technique, a solution containing the DNA fragment will be sequenced over the DNA subsequence chip. The subsequences of the DNA strand will hybridize the complementary probes (A-T and G-C are complementary to each other) in the DNA subsequence chip. The hybridized subsequences are given here in a random sequence to show the order information of the subsequences will not be available from the hybridization experiment. However, the main problem in hybridization technique is to reconstruct the target DNA sequence from the information available from the hybridized subsequences.

3.3 Application of the Stages of the Genetic Algorithm

Genetic algorithm is a search algorithm, which simulates natural evolution in optimization and search. Sequencing of DNA molecule involves massive computation in reconstructing the original DNA sequence. It is an optimization search technique in the domain of different sequences that can be constructed with the hybridized sequences. So, application of genetic algorithm is quite justified in the DNA sequencing problem. Our proposed method for the DNA sequencing problem through oligonucleotide hybridization based on genetic algorithm includes basically five stages; which are discussed below.

3.3.1 The Population Construction

Here we construct two populations from the hybridized probes sequences of length *N* and the subsequences of length *N+1*. Each population consists of chromosomes of fixed length formed from the subsequences and the populations try to reach the desired chromosome length. For example, TAGG can be connected to AGCC to form TAGGC. This procedure goes on until the length of the chromosome does not cross the chosen length.

For a given sequence ATGCATGA GTCA,
The length of the input sequence is *= 12,*
The length of first population is *= 8, i.e., N=8,*
Then length of the second population is *= 9, i.e., N+1 = 9.*

3.3.2 Fitness Function

The theory given by M. Calvino *et al.* represents one sequence to a certain subsequence of the population in which the individual differentiate themselves at most in the position of the asterisks [6].
For example, the subsequence H = GT*TG* correspond with the DNA sequence group {GTATGC, GTCTGA}. At the same time any subsequence group can be defined.

Let, *S* = number of elements present in the list.

Now, to define the subsequence as H_i; (*i*, represents the total number of subsequences present in the list)
$H_i = 0$, when all are G,
$= 1$, when all are T,
$= (0, 1),$ for other combinations.

To define subsequences of our proposed method, we have chosen four characters as a subsequence.
For example, a sequence GTACGTCAGTAC and the subsequences,
H_i= {GTAC, TACG, ACGT, CGTC, GTCA, TCAG, CAGT, AGTA, GTAC}.
When $i = 0, 1, 2, \ldots\ldots, 9$; the subsequences GTAC, TACG, ACGT, ……, GTAC will be selected respectively.
Where, $H_i = 0$, when any subsequence only once present in the list means the subsequence in the H_i will be once.
$H_i = 1$, when any subsequences present more than once.

So, we have the sub-sequence GTAC in the above set twice (more than once) means $H_i = 1$ and the other subsequences in the set are once means $H_i = 0.$

According to our proposed method we take two combinations of *0* and *1* for each subsequence in the set. During the reconstruction of a given sequence, the subsequence is selected with a probability proportional to fitness function, which is described by M. Calvino *et al.* [6].

$$p_i = f_i(s) / \sum f_i(s)$$

Where,

$$f_i(s) = A_i[j] / \sum_{i=1}^{n1} A_i[j]$$

$$\sum f_i(s) = \sum_{j=1}^{2(n-1)} (f_i(s))$$

For each subsequence of length *N,* we associated a *2(N-1*) dimensional vector (we will always have the binary value of each element of the vector by taking the reminder value of 2 for each element of the vector). If the vector is denoted as A_i then, $A_j[j]$ denotes the j^{th} element of the vector A_i, *n* is the length of each subsequence, and *n1* is the total number of subsequences in the list.

In case of repetition of subsequences the performance of the function decreases. To eliminate these repeated subsequences, we use hope function denoted as *E,* which is described by M. Calvino *et al.* [6].

$E = (P_i + H_i)$

When $H_i = 0$; then the value of P_i will be the fitness value and when $H_i = 1$; then the best value of *E* will be consider as fitness value.

DNA sequences have repetitive subsequences. To detect these repetitive units, we have introduced the above fitness function for the subsequences. Suppose for the sub sequence GTAC is matched each individual character with the main set of other subsequences. From the first one, two and three characters are matched with the main sequence set in the same way from the last one it is matched. The summation of the matching unit is the total number of fitness value, P_i of the sequence. For example, AGCC has been repeated consecutively in a sequence (that is the sequence contains AGCCAGCC), then the subsequences AGCC, GCCA, CCAG, CAGC will be also occur in the set. If the subsequence is found then only AGCC has a double consecutive repetition in the sequence.

3.3.3 Selection Operation

We use the roulette wheel selection method to deal with the selection operation [3]. We assign areas to every chromosome, where the chromosome with the highest fitness value has the largest area. The system randomly generates a value between zero and one. If the fitness value of the i^{th} chromosome in a population is P_i and P_t is the summation of the fitness values of all chromosomes, then the selected probability of the chromosome is denoted by P_i/ P_t. When the selection operation is performed, the system chooses a chromosome for a population according to its probability. Therefore, the larger the selected probability of a chromosome is, the more the opportunity of the chromosome to be chosen for performing the crossover operation.

3.3.4 Crossover Operation

System performs the crossover operation of chromosomes as described by Carlos A. Brizuela *et al.* [9] must choose a pair of chromosomes from a population by the selection operation. After a pair of chromosomes has been chosen by the selection operation, the system randomly generates a value between zero and one and compares it with a predefined crossover rate [0, 1] determined by the user. The crossover operation is constructed with the following ways for each population. Two chromosomes are matched to find common subsequences of length *N-1*. If more than one common site is found, then any one of them is randomly selected. Then both the chromosomes exchange their nucleotides.

3.3.5 Mutation Operation

After performing the crossover operation, the mutation operation is performed for each population. The system randomly generates a value between zero and one. Then, the system compares the randomly generated value with a predefined mutation rate [0, 1] determined by the user. A random point is selected in the chromosome of the subsequence. The higher the fitness values the higher the probability to crossover. This point breaks up the chromosome into two parts. Then, two arbitrary lengths are set for two broken fragments. Then the set of solution of subsequences are searched, so that the length of each broken segment can be increased by joining subsequences from the set using connection rule until the length of the string does not cross the chosen length. Then the fragment which has more fitness value is chosen as the mutated replacement of the original chromosome.

3.4 Generating DNA Sequence with Successor Choice

Three types of errors can be present in the spectrum [2]: negative ones, missing subsequence in the spectrum; positive ones, being erroneous subsequence and the subsequences can have both positive and negative errors.
Example: Let, the true target sequence is: T = TACTTAGTTGGCAGTTCAAA

Then the true spectrum S = {

sequence

(0) TACT
(1) ACTT
(2) CTTA
(3) TTAG
(4) TAGT
(5) AGTT
(6) GTTG
(7) TTGG
(8) TGGC
(9) GGCA

(10) GCAG
(11) CAGT
(12) AGTT
(13) GTTC
(14) TTCA
(15) TCAA
(16) CAAA

}

Fig. 2: Subsequences of the input sequence

If $S^{/}$ is the erroneous spectrum than we have, $S^{/}$ = {

(0) TACT
(1) ACTT
(2) CTTA
(3) TTAG
(4) TAGT
(5) AGTT
(6) GTAC
(7) TTGG
(8) TGGC
(9) GGCA
(10) GCAG
. **ATCG**
(11) CAGT
(12) AGTT
(13) ~~GACG~~
(14) TTCA
(15) TCAA
(16) CAAA

}

Fig. 3: Subsequences of the input sequence with errors

Here bold-faced subsequence (subsequence 6) is the positive error, a dot with bold-faced subsequence (subsequence between 10 and 11) is the negative error and bold-faced strike through subsequence (subsequence 13) represents both positive and negative error simultaneously.

In DNA sequence reconstruction procedure, successor choice is one of the main key points. We use fitness function in our proposed method to choose the appropriate successor for reconstruction, which gives the error free subsequences. In the following way we can choose the best successor as well as error free subsequence.

1. We have the input DNA sequence of *N* length, which is obtained from the biochemical phase of the isothermic SBH.
2. Derive a set of spectrum *S* of strings of equal length (here 4) from the original input sequence by shifting the characters one by one, which is called the subsequences.
3. Modify randomly the set by hybridization.
4. Select error free subsequences with the proposed fitness function, which will be the best choice of successor.

If the subsequence is followed by only one immediate subsequence, the correct one, then in the reduction phase error part replaces a positive error. We will choice the appropriate subsequence by using the fitness function of our proposed method. A threshold value is used to select the next free successor. The subsequence, which has higher threshold value, has priority for selection. Again, if there is subsequence, which is a negative error, then a threshold value will act to the error and we can choose the next successor to find the error free subsequence.

Now, we will explain how proper subsequences will be selected for a given original sequence. For each subsequence of length *N,* we associated a *2(N-1*) dimensional vector as mentioned in [6]. We will match the first letter of any randomly taken subsequence with the end of all the subsequences of the original sequence. Then the first two letters will be matched with the end of all the subsequences of the original sequence. Next first three letters will be then matched with the end of all the subsequences of the original sequence. Again the last three letters will be matched with the beginning of all the subsequences of the original sequence. The last two letters will be matched with the beginning of all the subsequences of the original sequence. Lastly the last letter will be matched with the beginning of all the subsequences of the original sequence. Hence corresponding vector will be constructed. From our proposed fitness function we can get the fitness value of each subsequence. And the highest value will be selected for the next operation in the genetic algorithm. For example*,* we have subsequences GTAG, TAGC, GCCG, CCGT, CGTA, AGCC, AGCC, TAGG, GCGT and GGCG. We will refer to these subsequences rather than the original subsequences for clarity of understanding. Now let us take the subsequence AGCC. The first letter of AGCC, A occurs in the end of other subsequences once (CGTA). The first two letters of AGCC, AG occur in the end of other subsequences only once (GTAG). The first three letters of AGCC, AGC occur in the end of other subsequences only once (TAGC). The last three letters of AGCC, GCC occur in the beginning of other subsequences only once (GCCG). The last two letters of AGCC, CC never occur in the beginning of any other sub-sequence. The last letter of AGCC, C occurs in the beginning of other subsequences only once (CCGT). Hence corresponding vector can be constructed as [1 1 1 1 0 1]. We will always have the binary value of each element of the vector by taking the reminder value of 2 for each element of the vector. In this way we can get the corresponding vectors for each subsequence. We will get the fitness value of each subsequence. The highest fitness value will be the best next successor to choose, which will remove erroneous subsequences.

In Fig. 3, we have positive error (subsequence 6). There the original subsequence is "GTTG" and the erroneous subsequence is "GTAC". So, according to our proposed method the erroneous part "AC" will be replaced by "TG".

Here we first have to recognize the “GTTG” present in the spectrum once or more than once. If the subsequence is once then from our proposed fitness function, $H_i = 0$.

By applying the function, $P_6 = f_6$ (subsequence) / $\sum_{0\text{-}16}$ (subsequences)
Here *0 - 16* (length of the spectrum) means we have to calculate the value of all the subsequences presented in the spectrum, which is *0* to *16* for the above example.

	0	……….	6	………….	16	
$S =$	(0)TACT	……………	(6)GTTG	………..	(16)CAAA	Value of $i = 6$

Fig. 4: Error free spectrum *S*

	0	………..	6	………..	16	
$S^{\surd}$ =	(0)TACT	……………	(6)GTAC	………..	(16)CAAA	Value of $i = 6$

Fig. 5: Erroneous spectrum $S^{\surd}$

Again if we have the subsequences more than once in the spectrum (there is no repeated subsequences in the spectrum *S*), then we will apply the hope function $E = (P_i + H_i)$ to choose the best successor.

Suppose after hybridization process we have the spectrum S = {(0) TACT (1) ACTT……….. (6) GTTG …..(9) TACT…….. (15) ACTT…. (20) CAAA }. Here, H_i = 1. We have repeated subsequences “TACT and ACTT”. So, repeated subsequences we will calculate the best hope function value to choose the successors.

$S =$	(0)TACT	(1) ACTT	……	(6)GTTG	……	(9) TACT	………	(15)ACTT

(0) TACT → (9) TACT (1) ACTT → (15) ACTT

Fig. 6: Duplication of subsequences in the spectrum

We have the negative error (subsequences between 10 and 11) in the Fig. 6; ATGC is not present in the original spectrum. We can find the next successor by applying our proposed method as we apply for positive error. In two existing approaches: a tabu-search method described by J. Blazewicz *et al.* [1] and a hybrid genetic algorithm described by J. Blazewicz *et al.* [6], the next successor was chosen randomly, which may be a wrong choice. But our proposed method can give the next best successor to be chosen. The key point that we described before is to find out the best successor, which is the subsequence with no error can be properly found out by our proposed method.

4.0 EXPERIMENTAL RESULTS AND DISCUSSION

At first we took the input of original DNA sequence of length 30, population size of length 20, crossover rate for the first population 0.7, crossover rate for the second population 0.9, mutation rate 0.2 and generation of 10. Then we executed our proposed method 50 times. Again we performed the same operation for the spectrum size of 50 individuals (the four nucleotide bases) with population size of length 20, crossover rate for the first population 0.7, crossover rate for the second population 0.7, mutation rate 0.2 and generation of 25; 70 individuals with population size of length 20, crossover rate for the first population 0.7, crossover rate for the second population 0.9, mutation rate 0.2 and generation of 35, and 90 individuals with population size of length 20, crossover rate for the first population 0.7, crossover rate for the second population 0.9, mutation rate 0.2 and generation of 40. For all the input we calculate the average optimal values in Table 1.

Table 1: Average optimal values of the propose method

Spectrum size	Optimal values
30	100%
50	100%
70	99.9%
90	99.9 %

Fig. 7 shows the optimal result of our method. First column corresponds to the input spectrum size of 30 individuals and here we get the exact output of the input sequence. In second, third and fourth column correspond to the input and output spectrum of size 50, 70 and 90 individuals (the four nucleotide bases) respectively.

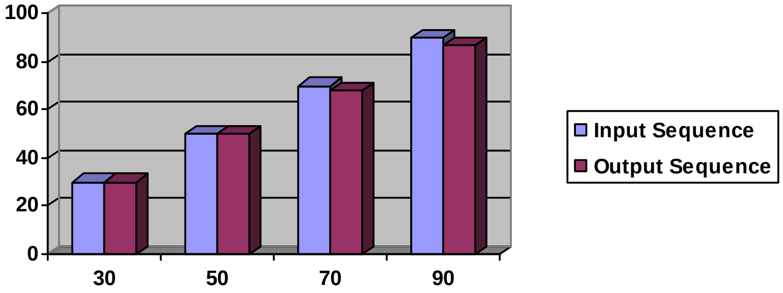

Fig. 7: Average optimal results of our method

Using the results of the computational experiments, the proposed method has been compared with the three other approaches: a tabu-search method described by J. Blazewicz *et al*. [1], a hybrid genetic algorithm described by J. Blazewicz *et al*. [7] and Sequencing by hybridization: an enhanced crossover operator for a hybrid genetic algorithm described by Carlos A. Brizuela *et al.* [9]. Here we denote tabu and scatter search as TSS, sequencing by hybridization as SBH, and hybrid genetic algorithm as HGA.

The experiment is performed on a PC with a Pentium 4, 1700 MHz, 256 MB RAM, and the Windows XP operating system. We use visual C++ language for performing this experiment. Table 2 shows the results of tests done with the tabu-search algorithm, the hybrid genetic algorithm, sequencing by hybridization, and our proposed method.

The spectrum size is 100 to 500 individuals (the four nucleotide bases). The numbers of optimal solutions returned by the algorithm; out of 40 are shown in the solutions that were construct using the optimal number if oligonucleotides (optimal quality). This does not necessarily using the optimal number of oligonucleotides generate the original sequence for the input sequence of 100, 200, 300, 400, and 500 lengths. The obtained solutions have the qualities that range from 99.1% to 100% of the optimal values (similarities between output sequences with original input sequences) of TSS algorithm, from 98.3% to 100% of the optimal values of HGA and from 99.1% to 99.75% SBH. As our proposed method can choice the best successors, we have the optimal values from 99.5% to 100%. From a practical standpoint, it is highly important that the proposed method returns many more optimal solutions that the previous methods. It can happen that a biochemical user, who would like to get a sequence reconstructed on the basis of his experiment, is interested only in obtaining the exact solution. Because the DNA sequencing problem with errors is highly complex this would normally be impossible using exact, exponential time algorithm. Thus, a method that runs in polynomial time, and that often returns optimal solutions, is very valuable. In the experiment, almost all solutions being optimal as measured by the global criterion function appear to be optimal also for biochemists, because they are identical to the original sequences that provide the data (something missing up to three nucleotides at the end because of negative errors).

Table 2: Optimal number comparisons

Spectrum size	**TSS (Tabu and Scatter Search)**	**SBH (Sequencing By Hybridization)**	**HGA (Hybrid Genetic Algorithm)**	**Proposed method**
100	40	39	40	40
200	38	37	38	38
300	31	31	20	32
400	21	19	9	22
500	18	15	5	20

All spectra used in the experiment are derived from real DNA sequences coding human proteins their accession numbers. The following idea has been used to introduce errors into the spectrum. The spectra have been sorted alphabetically and the first oligonucleotide of each original sequence has been known. We reduce both positive and negative errors and our method can also choice the best successors for reconstruct the original sequence. Our method shows better performance for long input sequence but takes more computational time compare to other method. The latter assumption is justified by information coming from biological experiments. Besides the three methods we present the results of tests done with our new method in the Tables 1 and 2. The results are much better than those of three existing methods. Moreover this method can give the idea of successor choice for a given sequence.

5.0 CONCLUSION AND SCOPE OF FUTURE WORK

We have presented an effective and efficient method for DNA sequence reconstruction based on genetic algorithm. We introduced successor choice operation to improve the solution quality. Experimental results showed that the method can handle a long input sequence, yielding very high quality output sequence. Experiments are performed to compare our proposed method with three other methods, a tabu and scatter search combined method, hybrid genetic approach, and sequencing by hybridization. In this paper we have presented the sequence length 100 to 500 individuals. We have better qualities of the optimal values than other existing methods. Reconstruction of DNA sequences, as a part of a wide stream of activities aiming to complete our knowledge about biochemical basis of life, attracts researchers from various domains.

In the method we have implemented DNA sequence reconstruction considering both positive and negative errors. But this method does not use any additional information about original sequence, which could be derived from the biochemical experiments and may help to obtain better similarity scores. We solved the problem of negative and positive errors as well as the problem of successor choice. That means, our method reconstruct the original sequence successfully. Moreover this method can produce better performance in relatively long sequence but it takes relatively more computational time. So, future work should be done on reducing the computational time.

REFERENCES

[1] Jacek Blazewicz, Fred Glover, Marta Kasprzak. *DNA Sequencing-Tabu and Scatter Search Combined*. Information Journal on Computing Vol. 16, No. 3, summer 2004, pp. 232-240.

[2] Md. Rafiqul Islam, Md. Shams-Ur Rahim, A. H. M. Saiful Islam and Mr. Shahidul Islam. *Reconstruction of a DNA sequence from its Probes in Sequencing by Hybridization*. International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2003.

[3] Shyi-Ming Chen*, Chung-Hui Lin, and Shi-Jay Chen. *Multiple DNA Sequence Alignment Based on Genetic Algorithms and Divide-and-Conquer Techniques*. International Journal of Applied Science and Engineering, 2005. 3, 2: 89-100.

[4] Md. Kamruzzaman, Sk. Mamunur Rashid, Arpita Majumder. *DNA Sequence Reconstruction Using Sequence*

By Hybridization Method With Negative And Positive Errors. Undergraduate thesis, CSE Discipline, Khulna University, Khulna, Bangladesh, 2005.

[5] Nicholas M. Hann, Simon J. Godsill. *Baysian Models for DNA Sequence*. Proceedings of IEEE international conference on acoustics, speech, and signal processing, 2002.0-7803-7402-9/02.

[6] Maria Calvino, Nuria Gomez, Luis F. Mingo. *DNA Simulation of Genetic Algorithm: Fitness Computation*, International Journal “Information Theories & Application”, Vol. 14 / 2007.

[7] Jacek Blazewicz, Piotr Formanowicz, Frederic Guinand and Marta Kasprzak. *A heuristic managing error for DNA sequencing*, Journal on bioinformatics vol. 18 no. 5, 2002, Pages 652-660.

[8] www.awl-he.con/biology

[9] Carlos A. Brizuela, Luis C. Gonzalez-Gurrol, Andrei Tchernykh, Denis Trystram, Sequencing by hybridization: an enhanced crossover operator for a hybrid genetic algorithm. J. Heuristics (2007) 13: 209–225.

[10] Jacek Blazewicz, Ceyda Oguz, Aleksandra Swiercz, *DNA Sequencing by Hybridization via Genetic Search*, OPERATION RESEARCH Vol.54, No. 6, November-December 2006, pp. 1185-1192.

[11] Jacek Blazewicz, Marta Kasprzakl, and Wojciech Kuroczycki, *Hybrid Genetic Algorithm for DNA Sequencing with Errors*, Journal of Heuristics, Vol. Number 5/ September, 2002, 495-502.

[12] J. Dylan Spalding, Cara MacNish, *A Genetic Algorithm to Sequence DNA using Sequencing by Hybridization Experimental Data*, Journal of Heuristics, 0-7803-7804-0/03/ 2003.

BIOGRAPHY

Md. Rafiqul Islam received M. Sc. in Engineering (Computers) from Azerbaijan Polytechnic Institute (at present Azerbaijan Technical University) in 1987 and PhD in Computer Science from Universiti Teknologi Malaysia (UTM) in 1999. Currently he is a Professor and Head of Computer Science and Engineering Discipline and Dean of Science, Engineering and Technology School of Khulna University, Bangladesh. He had published about 50 papers in national and international journals as well as international conference proceedings. His areas of interest include design and analysis of algorithms in the area of information security, external sorting, data compression, bio-informatics, grid computing, information retrieval etc.

Md. Rowshan Shahriar received B.Sc. in Computer Science and Engineering (CSE) from Khulna University, Bangladesh in 2007. His areas of interest include networking, genetic algorithm, computer security, database system and distributed system.

Abul Faisal Mohammad Shaheed received B.Sc. in Computer Science and Engineering (CSE) from Khulna University, Bangladesh in 2007. His areas of interest include genetic algorithm, computer architecture, networking, distributed systems and operating system.