GPCODON ALIGNMENT: A GLOBAL PAIRWISE CODON BASED SEQUENCE ALIGNMENT APPROACH

Zeinab A. Fareed[1], Hoda M. O. Mokhtar[1], and Ahmed Ahmed [2]

[1] Faculty of Computers and Information, Cairo University, Cairo, Egypt
[2] Independent Researcher, Luxembourg.

ABSTRACT

The alignment of two DNA sequences is a basic step in the analysis of biological data. Sequencing a long DNA sequence is one of the most interesting problems in bioinformatics. Several techniques have been developed to solve this sequence alignment problem like dynamic programming and heuristic algorithms. In this paper, we introduce (GPCodon alignment) a pairwise DNA-DNA method for global sequence alignment that improves the accuracy of pairwise sequence alignment. We use a new scoring matrix to produce the final alignment called the empirical codon substitution matrix. Using this matrix in our technique enabled the discovery of new relationships between sequences that could not be discovered using traditional matrices. In addition, we present experimental results that show the performance of the proposed technique over eleven datasets of average length of 2967 bps. We compared the efficiency and accuracy of our techniques against a comparable tool called "Pairwise Align Codons" [1].

KEYWORDS

DNA, Sequence alignment, Pairwise alignment, Global Alignment.

1. INTRODUCTION

Global sequence alignment is one of the most challenging tasks in bioinformatics. There are many global alignment techniques that have been applied in biology. Nevertheless, the alignment of whole genome is still a problem in bioinformatics due to their large sizes that require extensive computations. Thus, efficient and accurate DNA sequence alignment techniques are needed. Using traditional pairwise sequence alignment is hence infeasible, more efficient approaches are needed to efficiently handle whole DNA alignment.

Some tools can be used to achieve good execution time by aligning the well matched segments that can be joined together using dynamic programming technique to find the alignment. For example Delcher et al [2] used a suffix tree to find the best match of a given length called MUMs. Many sequence alignment techniques have been developed, specially, for string matching. Some are based on dynamic programming like Needleman-Wunsch [3], BLAST [4], FASTA [5], others are based on statistical methods like MUMmer [2], AVID [6], LAGAN [7].

Several software tools were also developed to find the similarity between biological sequences. The commonly used tool for local alignment is BLAST [4]. Another software tool for multiple alignment that combines both local and global alignment is DIALIGN, this tool uses dynamic programming [8].

The main contribution in this work is introducing a new accurate and efficient method to achieve better alignment with high score in less time, this method is based on employing a new scoring matrix known as "Empirical Codon Substitution Matrix" [13], and we refer to our proposed approach as *GPCodon alignment.*

DOI : 10.5121/ijdms.2016.8101

The rest of the paper is organized as follows: section 2 overviews some basic points about sequence alignment and its techniques. Section 3 presents an overview about bioinformatics algorithms in previous works. While section 4 presents our proposed GPCodon alignment approach. Section 5 illustrates the experimental results and analysis of the work using di_erent test cases is well illustrated. Finally the discussion and conclusion is demonstrated in section 7 as well as directions for future work.

2. BACKGROUND

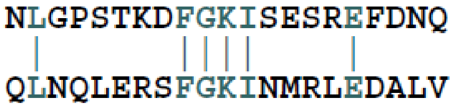
In this section, we discuss some of the basic points that will be used later in the proposed approach. We will mainly focus on three main concepts, namely: sequence alignment, scoring matrices, and codon sequence alignment.

2.1. Sequence Alignment

Sequence alignment is an approach to retrieve the best match between two or more sequences. The most important factor in sequence alignment is choosing the scoring schema will be employed, choosing a bad scoring schema will lead to inaccurate alignments [9]. There are two types of sequence alignment global alignment and local alignment.

2.1.1. Global Alignment

Global alignment assumes that we have two sequences which are basically similar over the whole length of one another. The alignment aims to match them to each other from end to end, although parts of the alignment are a bit different [8].

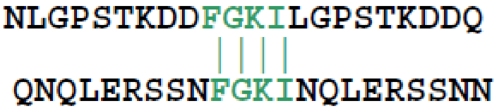


```
NLGPSTKDFGKISESREFDNQ
 |      ||||    |
QLNQLERSFGKINMRLEDALV
```

Some global alignment techniques introduce gaps into the sequences for the purpose of increasing the overall alignment score. Nevertheless, introducing a gap adds a penalty to the score but might enhance the overall score.

2.1.2. Local Alignment

Local alignment searches for some regions of the two sequences that match well. There is no aim to force entire sequences into an alignment, just those parts that have good similarity. Using the same sequences as above, local alignment becomes as follow: [8].



```
NLGPSTKDDFGKILGPSTKDDQ
         ||||
QNQLERSSNFGKINQLERSSNN
```

Similarly, gaps could be introduced with penalty if it increases overall score.

2.2. Scoring Matrices

Generally, to align two DNA sequences, a score is given to a matched or mismatched pairs of nucleotides [17]. A scoring matrix is used to measure the degree of similarity between sequences, this can be used in both local and global alignment. To build this matrix an appropriate scoring function should be used to favor the matched nucleotides and penalize the unmatched nucleotides. The most popular scoring matrices are: Point Accepted Mutation matrix (PAM) [11], Blocks Substitution Matrix (BLOSUM) [12] and Empirical Codon Substitution Matrix [13]. In the following discussion we will explore the Empirical Codon Substitution Matrix as it is the matrix selected for the presented work.

2.2.1. Empirical Codon Substitution Matrix

Empirical Codon Substitution Matrix [13] is used to score sequence alignments. Empirical Codon Substitution matrix was introduced by Adrian Schneider, Gina M Cannarozzi and Gaston H Gonnet in 2005 based on 17,502 alignments. It's the first scoring matrix built from alignments of DNA sequences, it describes the substitutions probabilities for each codon for a specific evolutionary distance.

A higher score in the matrix means that this transition is more similar than one with a lower score. The matrix is symmetric, i.e. the score from codon i to j have the same score as from j to i. The matrix is built from pairwise alignments of sequences from 5 species, human (Homo sapiens), mouse (Mus musculus), chicken (Gallus gallus), frog (Xenopus tropicalis) and zebrafish (Brachydanio rerio) [13]. One of the applications that uses the Empirical Codon Substitution Matrix in its algorithm to globally score the alignment between pair of sequences is Pairwise Align Codons [1]

2.3. Pairwise Align Codons

Pairwise Align Codons is an online pairwise sequencing tool. It takes two DNA sequences and determines the global alignment between them. The used scoring matrix in this tool to calculate the alignment is the Empirical Codon Substitution matrix. Despite the value of this tool, it suffers from the following limitations.

1- Each sequence of the submitted sequences should be divisible by 3.
2- Length of each sequence should be less than or equal 6000 bps.

3. RELATED WORKS

There are some techniques that have been proposed for sequence alignment including the work in [15], [16], and [13]. In [15] the authors proposed a program called "BLAST" which can be used for searching DNA and protein databases for sequence similarities. It compares protein or DNA queries with protein or DNA databases. In [3], The Basic Local Alignment Search Tool (BLAST) is used to explore the regions of local similarity between sequences. It compares DNA or protein sequences to sequence databases, and then computes the percentage of matching between them. Also, it can be used for determining the functional and evolutionary relationships among sequences to identify members of gene families.

In [16] the authors updated the algorithm of genome sequence alignment which is called EDAGSA. In the paper, the authors presented an algorithm where only the entire three main diagonals are scored without filling the whole matrix with unused data. In [13], the authors presented the first empirical codon matrix which is built from coding sequences from vertebrate DNA sequences. In [8] the authors proposed a method for sequence alignment that is based on index pattern matching using multi-threading, this index has been used to obtain an optimal alignment score.

In [9], they proposed a technique called “gpALIGNERr”, he used “spaced seeds” to locally aligned subsequences and used the same scoring function with DIALIGN-T to produce the final alignment. In [17], an algorithm called PROVEAN (Protein Variation Effect Analyzer) was proposed, the proposed approach is a metric to compute the functional effect of variations, and it can be naturally applied to any variations of protein sequence. In [18], the authors presented a technique called FOGSAA that employs the famous branch and bound technique and is based on global pairwise sequence alignment. It builds a branch and bound tree where each node represents a comparison between two letters and each path represents a sequence alignment between two sequences.

In [22], a new algorithm was proposed that can be used to find exact occurrences of patterns in DNA sequences, it uses a matching pattern technique called “An Index Based Pattern Matching using Multithreading”, this technique can be used for pattern matching in protein sequences and for English text as well. In [4], the authors presented an algorithm that can be used to search for similarities between protein sequences, the algorithm firstly identifies regions of similar sequence and then scores the identical residues in those regions.

In [5], the authors modified the Longest Common Subsequence algorithm to be Fast Longest Common Subsequences (FLCS). The basic point in these modifications is to ignore the unused data of the Longest Common Subsequences matrix and evaluate only the three main diagonals of the FLCS matrix. In [23], the authors proposed a system for aligning and comparing whole genomes called “MUMmer”. It can be used for aligning two sequences. It can be also used for aligning pair of sequences.

4. GPCODON ALIGNMENT

In this paper we propose a new computational approach that determines the pairwise alignment between two sequences. Our proposed method is a modified global pairwise alignment that accepts two coding sequences and determines the optimal global alignment. The scoring matrix that the proposed method uses is the Empirical Codon Substitution matrix.

Algorithm 1 Building the calculation matrix between sequences X and Y

```
//initialization
for $i = 0$ to $4$
    for $j = 0$ to $m + 2$
        $Matrix(i,j) \leftarrow 0$
    end for
end for
forj $= 0$ to $4$
    for $i = 0$ to $n + 2$
        $Matrix(i,j) \leftarrow 0$
    end for
endfor
for $i = 5$ to $m + 2$
    for $j = 5$ to $n + 2$
        //recursionsteps
        if $Codon1 = Codon2$
            $Matrix(i,j) \leftarrow Max(Match, Delete, Insert)$
```

```
$Match = Max\left\{\begin{matrix} Matrix(i-3,j-3) \\ Matrix(i-3,j-4) \\ Matrix(i-3,j-5) \\ Matrix(i-4,j-3) \\ Matrix(i-4,j-4) \\ Matrix(i-4,j-5) \\ Matrix(i-5,j-3) \\ Matrix(i-5,j-4) \\ Matrix(i-5,j-5). \end{matrix} \quad +S(Xi,Yj)\right\}$

$S(Xi,Yj)$ is the score between these codons in the empirical codon substitution matrix.

$Delete = Max\left\{\begin{matrix} Matrix(i-3,j) \\ Matrix(i-4,j) \\ Matrix(i-5,j). \end{matrix}\right\}$
$Insert = Max\left\{\begin{matrix} Matrix(i,j-3) \\ Matrix(i,j-4) \\ Matrix(i,j-5). \end{matrix}\right\}$
else
   $Matrix(i,j) \leftarrow Max(Delete, Insert)$
   $Delete = Max\left\{\begin{matrix} MATRIX(i-3,j) \\ MATRIX(i-4,j) \\ MATRIX(i-5,j). \end{matrix}\right\}$
end if
endfor
endfor
if $(Matrix(i,j) = Match)$
   $Directions(i,j).Sign = 2$
   $Directions(i,j).positioni = positioni$ of the maximum Diagonal score
   $Directions(i,j).positionj = positionj$ of the maximum Diagonal score
end if
if $(Matrix(i,j) = Insert)$
   $Directions(i,j).Sign = 1$
   $Directions(i,j).positioni = positioni$ of the maximum horizontal score
   $Directions(i,j).positionj = positionj$ of the maximum horizontal score
end if
if $(Matrix(i,j) = Delete)$
   $Directions(i,j).Sign = 0$
   $Directions(i,j).positioni = positioni$ ofthe maximum vertical score
   $Directions(i,j).positionj = positionj$ ofthe maximum vertical score
end if
```

Algorithm 1 computes the alignment score between two sequences *X* of length *n* and *Y* of length *m*. Here *Matrix (i,j)* stores the highest score between X_{1i} and Y_{1j} and *Directions (i,j)* keeps additional information about which of the quantities *Matrix(i-3,j), Matrix(i-4,j), Matrix(i-5,j)* corresponds to the maximum of *Matrix(i,j)*. Each score in *Matrix (i,j)* shows the similarity between 2 codons (trinucleotides), codon from *i* to *i+3* in *X* and codon from *j* to *j+3* in *Y*.

Algorithm 2 Backtracking

```
$Input: The\ calculation\ matrix.$
$//i, j\ stand\ for\ the\ position\ of\ the\ maximum\ score.$
$While\ i > 0\ or\ j > 0$
    $if\ (Matrix(i,j) = Match)$
        $Seq1 = Sequence1.substring(i-3, i) + Seq1.$
        $Seq2 = Sequence2.substring(j-3, j) + Seq2.$
        $i -= 3.$
        $j -= 3.$
    $end\ if$
    $if\ (Matrix(i,j) = Insert)$
        $Seq1 = --- + Seq1.$
        $Seq2 = Sequence2.substring(j-3, j) + Seq2.$
        $j -= 3.$
    $end\ if$
    $if\ (Matrix(i,j) = Delete)$
        $Seq1 = Sequence1.substring(i-3, i) + Seq1.$
        $Seq2 = --- + Seq2.$
        $i -= 3.$
    $end\ if$
```

Example 1:

Consider the Input sequences A and B as shown in Fig.1

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

Figure 1: Input Sequences A and B

Input: Given two sequences A, B of length m and n respectively, empirical codon scoring matrix.

Output: The score and an alignment of the two sequences such that all characters in both sequences should be participated.

Here the length of sequence A and B =13. If the length of sequences is longer than that we can subdivide these sequences into sub sequences to fit in the memory. The algorithm steps are as follows:

Step 1: [initialization of variables]

1 Set *Matrix(0, j)* = *Matrix (1, j)* = *Matrix (2, j)* = *Matrix (3, j)* = *Matrix (4, j)* = *Matrix(i,0)* = *Matrix (i,1)* = *Matrix (i,2)* = *Matrix (i,3)* = *Matrix (i,4)* = 0.

2 We have two cases to calculate each cell in the matrix, either both codons are matching or mismatching.

Matched codons

The score between two matched codons can be calculated by three ways. The first way is to choose the max score among nine cells from the diagonal, secondly by getting the max of three cells from the vertical path, and thirdly by choosing the max of three scores from the horizontal path as shown in Algorithm 1.

Mismatched codons:

We can get the score of these codons by calculating the max of horizontal and vertical scores as shown in Algorithm 1.

Step 2: [Main Iteration]

In this iteration, we calculate each cell in the matrix as shown in Figure 2.

<u>For example:</u>

Matrix [5, 5] can be calculated by two ways.

1- *Matrix[5,5] = max(Matrix[1,5],Matrix[2,5],Matrix[3,5])*= 0, *Dir.* = Vertical.

2- *Matrix* [5, 5] = *max* (*Matrix* [5, 1], *Matrix* [5, 2], *Matrix* [5, 3]) = 0, *Dir.* = Horizontal.

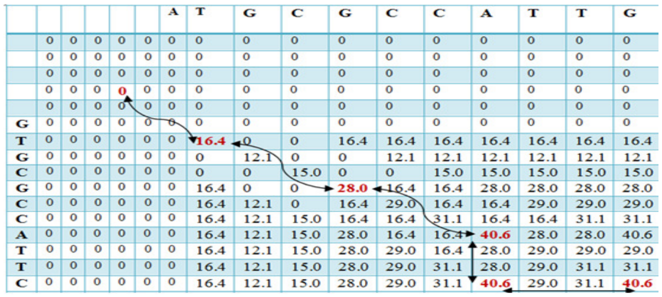
Matrix [6, 6] can be calculated by three ways.

1- *Matrix* [6, 6]= max(*Matrix* [1,1], *Matrix* [1,2], *Matrix* [1,3], *Matrix* [2,1], *Matrix* [2,2], *Matrix* [2,3], *Matrix* [3,1], *Matrix* [3,2], *Matrix* [3,3]) + corresponding score from the empirical matrix = 0 + 16.4 = 16.4, where The corresponding empirical score of (TGC, TGC) = 16.4, *Dir.* = diagonal.

2- *Matrix* [6, 6] = max (*Matrix* [1, 6], *Matrix* [2, 6], *Matrix* [3, 6]) = 0. *Dir.* = Vertical.

3- *Matrix* [6, 6] = max (*Matrix* [6, 1], *Matrix* [6, 2], *Matrix* [6, 3]) = 0. *Dir.* = Horizontal.

And so on for all cells, the values in the matrix will be as shown in Figure. 2. The directions array saves the path of the optimal alignment.

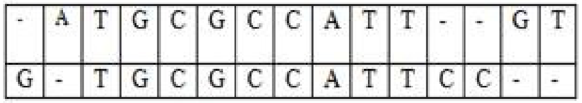


						A	T	G	C	G	C	C	A	T	T	G
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	16.4	0	0	16.4	16.4	16.4	16.4	16.4	16.4	16.4
G	0	0	0	0	0	0	0	12.1	0	0	12.1	12.1	12.1	12.1	12.1	12.1
C	0	0	0	0	0	0	0	0	15.0	0	0	15.0	15.0	15.0	15.0	15.0
G	0	0	0	0	0	0	16.4	0	0	28.0	16.4	16.4	28.0	28.0	28.0	28.0
C	0	0	0	0	0	0	16.4	12.1	0	16.4	29.0	16.4	16.4	29.0	29.0	29.0
C	0	0	0	0	0	0	16.4	12.1	15.0	16.4	16.4	31.1	16.4	16.4	31.1	31.1
A	0	0	0	0	0	0	16.4	12.1	15.0	28.0	16.4	16.4	40.6	28.0	28.0	40.6
T	0	0	0	0	0	0	16.4	12.1	15.0	28.0	29.0	16.4	28.0	29.0	29.0	29.0
T	0	0	0	0	0	0	16.4	12.1	15.0	28.0	29.0	31.1	28.0	29.0	31.1	31.1
C	0	0	0	0	0	0	16.4	12.1	15.0	28.0	29.0	31.1	40.6	29.0	31.1	40.6

Figure 2: Shows the best matches in the matrix with the trace back.

Step3: Termination

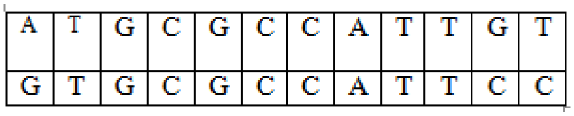
After finishing the calculation matrix, we choose the maximum score among the last nine scores from the calculation matrix which is 40.6 in our example. After finding the maximum score from the matrix, we can trace back to the optimal alignment. The final alignment is as shown below



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

Figure 3: Alignment of sequences A and B.

<u>Example 2</u>: Consider the Input sequences A and B as shown in Fig.4

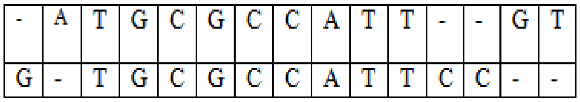


A	T	G	C	G	C	C	A	T	T	G	T
G	T	G	C	G	C	C	A	T	T	C	C

Figure 4: Input Sequences A and B

The length of sequence A and B =12. The alignment score of our algorithm is 40 by following the above steps to fill the calculation matrix. On the other hand, we got a score 30 after using the pairwise align codons(online tool). The difference between our algorithm and the pairwise align codons is the scoring function. In our algorithm, we are calculating the score for each cell by choosing the maximum score among three cells from vertical path and three cells from horizontal path and nine cells from diagonal path as we illustrated earlier in the algorithm, while the scoring function of pairwise align codons is based on choosing the maximum score from *Matrix [i - 1] [j], Matrix[i] [j - 1]* and *Matrix [i - 1] [j -1]* to fill each cell in the calculation matrix, That's why our algorithm is generating a higher score than the pairwise align codons.

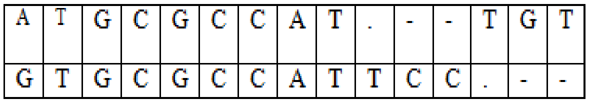
The alignment of our GPCodon alignment approach is as follows:



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

Figure 5: Alignment of our algorithm.

The alignment of pairwise align codons is shown below:



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

Figure 6: Alignment of the pairwise align codons

5. Case study and Interpreting Results

In the following discussion, we discuss the data and the statistical tools that have been used, and then we will show our experimental results for testing the proposed approach.

5.1. Sample Population

We performed our experiments based on multiple datasets from the complete genome databases and real sequences of NCBI database. In the comparison, we used 12 sequences, these sequences have been divided into datasets, and each dataset contains 2 sequences. Our experimental setting is as follows: we developed our methods using Java language and we conducted the experiments under the Windows OS on an Intel core i7 PC with RAM 1 GB.

5.2. Interpreting Results

In this section we present our experimental results to measure the efficiency of the proposed GPCodon method. We evaluated the absolute running time and the alignment score for each alignment using the GPCodon method and the online global alignment tool which is "Pairwise Align codons". The following table shows the running time and the accuracy of each approach for different cases.

Table 1. Execution time between proposed approach and pairwise align codons.

#	GenBank ID	Length (bp)	Algorithm	Time(Seconds)
1	HQ180395.1	1371	GPCodon alignment	3
	NC 024372.1	1374	Pairwise Align codons	.2
2	NC 026138.1	2223	GPCodon alignment	10
	NC 026261.1	2238	Pairwise Align codons	.5
3	NC 026163.1	2277	GPCodon alignment	10
	NC 027798.1	2292	Pairwise Align codons	.52
4	NC 020254.1	2664	GPCodon alignment	13
	NC 010797.1	2736	Pairwise Align codons	.72
5	NC 026270.1	3348	GPCodon alignment	24
	NC 002187.1	4014	Pairwise Align codons	1.58
6	NC 001600.1	4041	GPCodon alignment	34
	NC 002194.1	4563	Pairwise Align codons	2.21
7	NC 026163.1	1371	GPCodon alignment	5
	HQ180395.1	2277	Pairwise Align codons	.42
8	NC 027798.1	2292	GPCodon alignment	6
	NC 024372.1	1374	Pairwise Align codons	.40
9	NC 020254.1	2664	GPCodon alignment	18
	NC 002187.1	4014	Pairwise Align codons	1.31
10	NC 026270.1	3348	GPCodon alignment	17
	NC 010797.1	2736	Pairwise Align codons	1.11
11	NC 026138.1	4041	GPCodon alignment	33
	NC 002194.1	4563	Pairwise Align codons	2.22

Table 1 summarizes the execution time of the proposed approach compared to the pairwise align codons over different datasets, from that table we can conclude that the pairwise align codons is the fastest one.

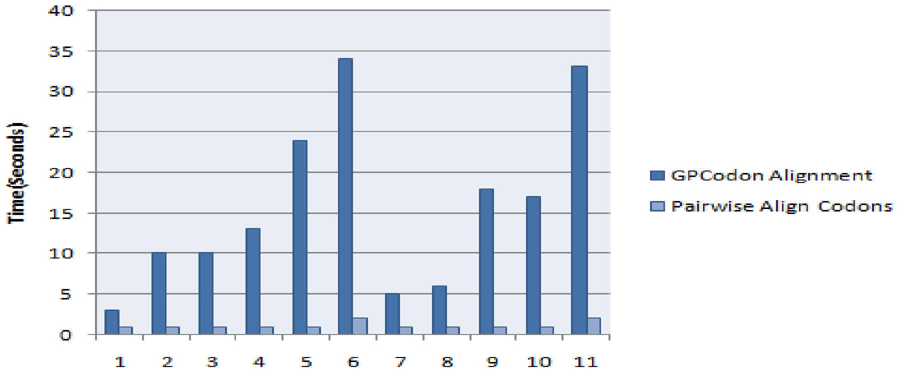


Figure 7. Execution time for the proposed approach and the online tool.

Fig. 7 shows the execution time of different datasets, each dataset contains two sequences of different lengths, these sizes range from 1371 to 4563 nucleotides. From the comparison we can found that Pairwise Align Codons presents least execution time compared with the proposed technique. In the following table, we show the accuracy for the proposed method using different sizes from 1371 to 4563 nucleotides.

Table 2. The alignment scores of the proposed approach and the pairwise align codons.

#	GenBank ID	Length (bp)	Algorithm	Score
1	HQ180395.1	1371	GPCodon alignment	3113
	NC 024372.1	1374	Pairwise Align codons	2101
2	NC 026138.1	2223	GPCodon alignment	4729
	NC 026261.1	2238	Pairwise Align codons	3163
3	NC 026163.1	2277	GPCodon alignment	5021
	NC 027798.1	2292	Pairwise Align codons	3100
4	NC 020254.1	2664	GPCodon alignment	6168
	NC 010797.1	2736	Pairwise Align codons	3905
5	NC 026270.1	3348	GPCodon alignment	7928
	NC 002187.1	4014	Pairwise Align codons	5075
6	NC 001600.1	4041	GPCodon alignment	9492
	NC 002194.1	4563	Pairwise Align codons	5861
7	NC 026163.1	1371	GPCodon alignment	3768.8
	HQ180395.1	2277	Pairwise Align codons	2497
8	NC 027798.1	2292	GPCodon alignment	3742
	NC 024372.1	1374	Pairwise Align codons	2563
9	NC 020254.1	2664	GPCodon alignment	7175
	NC 002187.1	4014	Pairwise Align codons	4808
10	NC 026270.1	3348	GPCodon alignment	6548
	NC 010797.1	2736	Pairwise Align codons	4201
11	NC 026138.1	4041	GPCodon alignment	9492
	NC 002194.1	4563	Pairwise Align codons	5861

As shown in Table 2, we present the accuracy for proposed approach using different dataset sizes from 1371 to 4563 nucleotides. From table II and Fig. 8, we can conclude that the proposed approach is more accurate than the pairwise align codons because the scores of the proposed approach are higher than scores of pairwise align codons.

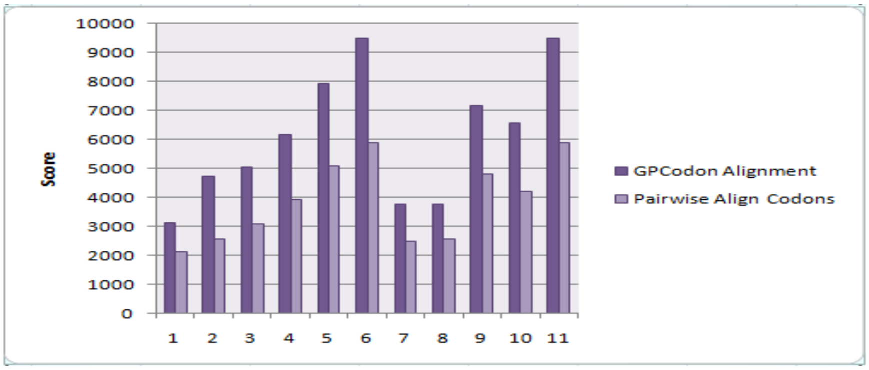


Figure 8. Accuracy of the proposed approach and the online tool.

6. DISCUSSION

From the test cases, we can find that the proposed method is more accurate than the pairwise align codons because while building Our calculation matrix, we check each pair of codons for matching, if we are comparing two matched codons then we will calculate it by selecting the maximum score among the nine scores from diagonal, vertical and horizontal scores as we illustrated in the algorithm, and then we use the equivalent score that measures the similarity between these codons from the empirical substitution matrix to be added to the selected maximum score. In case they are not equal, we should choose the appropriate max score from the vertical or horizontal scores and then we add the equivalent score of similarity between them. After finishing the scoring, we choose the maximum score among the last nine scores from the scoring matrix and finally we trace back to find the optimal alignment.

7. CONCLUSIONS AND FUTURE WORK

Today, due to the large size of DNA sequences, traditional sequence alignment tools are not feasible. To solve this issue, we should use an accurate and efficient sequence alignment method. In this study, we introduced a new pairwise sequence alignment method for finding the optimal alignment between two DNA sequences based on codons instead of nucleotides. The proposed method is based on a scoring matrix which is called “Empirical Codon Substitution matrix”. We carried out experiments on the proposed method using six datasets and the experiments showed the efficiency of the proposed technique. The experiments also illustrate an improvement in the running time and the alignment score. For future work, we plan to enhance the execution time of our method and investigate the effectiveness of running our proposed approach on larger datasets.

REFERENCES

[1] P. Stothard, http://www.bioinformatics.org/sms2/pairwise align codons.html, 2000.

[2] N. Bray, I. Dubchak, and L. Pachter, (2003)“Avid: A global alignment program,” Genome research, vol. 13, no. 1, pp. 97–102.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, (1990)“Basic local alignment search tool,” Journal of molecular biology, vol. 215, no. 3, pp. 403–410.

[4] D. J. Lipman and W. R. Pearson, (1985)“Rapid and sensitive protein similarity searches,” Science, vol. 227, no. 4693, pp. 1435–1441.

[5] M. Ossman and L. F. Hussein, (2012) “Fast longest common subsequences for bioinformatics dynamic programming,” population, vol. 5, p. 7.

[6] M. Brudno, M. Chapman, B. G¨ottgens, S. Batzoglou, and B. Morgenstern, (2003) “Fast and sensitive multiple alignment of large genomic sequences,” BMC bioinformatics, vol. 4, no. 1, p. 66.

[7] A. Dhraief, R. Issaoui, and A. Belghith, (2011) “Parallel computing the longest common subsequence (lcs) on gpus: efficiency and language suitability,” in The 1st International Conference on Advanced Communications and Computation (INFOCOMP).

[8] S. N. Devi and S. Rajagopalan, (2012) “A modified dynamic parallel algorithm for sequence alignment in biosequences,” International Journal of Computer Applications, vol. 60, no. 18.

[9] M. Hadian Dehkordi, A. Masoudi-Nejad, and M. Mohamad-Mouri, (2011) “gpaligner: A fast algorithm for global pairwise alignment of dna sequences,” Iran. J. Chem. Chem. Eng. Vol, vol. 30, no. 2.

[10] N. C. Jones and P. Pevzner, (2004) “An introduction to bioinformatics algorithms”. MIT press.

[11] M. O. Dayhoff and R. M. Schwartz, (1978) “A model of evolutionary change in proteins,” in In Atlas of protein sequence and structure. Citeseer.

[12] P. Stothard, (2000) “The sequence manipulation suite: Javascript programs for analyzing and formatting protein and dna sequences.” Biotechniques, vol. 28, no. 6, pp. 1102–1104.

[13] A. Schneider, G. M. Cannarozzi, and G. H. Gonnet, (2005) “Empirical codon substitution matrix,” BMC bioinformatics, vol. 6, no. 1, p. 134.

[14] J. Pevsner, (2005) Bioinformatics and functional genomics. John Wiley & Sons.

[15] S. F. Altschul, T. L. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, (1997) “Gapped blast and psiblast: A new generation of protein database search programs,” nucleic acids research, vol. 25, no. 17, p. 33893402.

[16] A. E. keshk, (2013) “Enhanced dynamic algorithm of genome sequence alignments,” International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 2, no. 6.

[17] Y. Choi, (2012) “A fast computation of pairwise sequence alignment scores between a protein and a set of single-locus variants of another protein,” in Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine. ACM, pp. 414–417.

[18] A. Chakraborty and S. Bandyopadhyay, (2013) “Fogsaa: Fast optimal global sequence alignment algorithm,” Scientific reports, vol. 3.

[19] G. W. Klau, (2009) “A new graph-based method for pairwise global network alignment,” BMC bioinformatics, vol. 10, no. Suppl 1, p. S59.

[20] T. F. Smith and M. S. Waterman, (1981) “Identification of common molecular sub sequences,” Journal of molecular biology, vol. 147, no. 1, pp. 195–197.

[21] S. B. Needleman and C. D. Wunsch, (1970) “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” Journal of molecular biology, vol. 48, no. 3, pp. 443–453.

[22] S. N. Devi and S. Rajagopalan, (2012) “An index based pattern matching using multithreading,” International Journal of Computer Applications, vol. 50, no. 6.

[23] A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg, (1999) “Alignment of whole genomes,” Nucleic Acids Research, vol. 27, no. 11, pp. 2369–2376.

[24] W. S. Martins, J. del Cuvillo, W. Cui, and G. R. Gao, (2001) “Whole genome alignment using a multithreaded parallel implementation,” in Symposium on Computer Architecture and High Performance Computing, pp. 1–8.

[25] J. Li, S. Ranka, and S. Sahni, (2014) “Pairwise sequence alignment for very long sequences on gpus,” International Journal of Bioinformatics Research and Applications 2, vol. 10, no. 4-5, pp. 345–368.

[26] R. C. Edgar, (2004) “Muscle: multiple sequence alignment with high accuracy and high throughput,” Nucleic acids research, vol. 32, no. 5, pp. 1792– 1797.

[27] S. Chen, C. Lin et al., “Multiple dna sequence alignment based on genetic simulated annealing techniques,” International journal of information and management sciences, vol. 18, no. 2, p. 97, 2007