Received: January 25, 2023. Revised: February 18, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Developing New Pairwise Sequence Alignment Method based on Needleman-Wunsch Algorithm

Rana H. Saloom[1]* **Hussein K. Khafaji[2]**

[1]Informatics Institute for Postgraduate Studies (IIPS),
Iraqi Commission for Computers and Informatics (ICCI) Baghdad, Iraq
[2]Computer Communications Engineering Department, Al-Rafidain University College Baghdad, Iraq
* Corresponding author's Email: Phd202020565@iips.icci.edu.iq

Abstract: Molecular biology and bioinformatics are fields that alter knowledge and abilities for gathering, managing, storing, analyzing, interpreting, and disseminating biological data. In order to get insight into the design and testing of life sciences, it is necessary to make use of high-performance computers, cutting-edge software tools, and unique algorithmic approaches for data analysis, interpretation, and prognostication. The computing challenges of processing biological sequences are used in this research to describe a significant technique for producing DNA sequence alignments. One of these challenges is to put forward computational models that aid in speeding up the computation of the substitution matrix of the needleman-wunsch algorithm. Another difficulty is the implementation of a parallel program for two threads to find the values of the matrix concurrently in order to achieve high efficiency and find the best sequence alignment in less time. This paper introduces a novel approach to constructing the substitution matrix of the Needleman-Winch algorithm. This approach enhances the overall performance of the algorithm by adjusting the values of the penalties imposed for the gap, mismatch, and match in addition to computing the matrix using two parallel threads depending on the progress made in the devices, especially with regard to the capabilities and performance of the GPU. Numerous tests have been run using different data sets and different lengths of sequence. A comparison was made between the amount of time needed to implement the Needleman method and the amount of time needed to implement the suggested model using the same data in order to evaluate the performance of the proposed model. Processing time and speedup for parallel performance are experimentally calculated. The recommended model has high scalability in terms of workload and machine capacity, according to the performance evaluation and scalability assessments. The proposed approach reduced the execution time ratio by 47% to 90%, and this improvement rate rises as the length of the reference and query sequences does.

Keywords: Bioinformatics, DNA sequences, Mutation, Alignment, Needleman-wunsch.

1. Introduction

An interdisciplinary field called bioinformatics combines science and programming to provide methods and algorithms for interpreting and understanding biological data. It developed out of the need for guidelines for the massive amounts of data that atomic physics generated, and it is today crucial to the fields of genetics and bioinformatics [1] .In the field of bioinformatics, biology-related topics are addressed using mathematical, computer science, and engineering approaches [2] .This may be accomplished by using a variety of strategies to frame and model the relevant biological problems as computational problems and create algorithms to address them precisely and effectively [3].

One of the topics that receives the most attention in bioinformatics is sequence alignment. It is involved with identifying patterns (similarities) inside a sequence or between sequences. This issue covers a broad range of techniques for locating sections of biological sequences that have a lot of similarities, such as those in DNA, RNA, and protein [4]. Several methods, including a heuristic method and a dynamic programming method, were developed for sequence alignment at the cost of precision [5]. The Needleman-Wunsch (NW) method for the global alignment and the Smith-Waterman (SW) algorithm for the local alignment are two well-known dynamic programming-based pairwise sequence alignment techniques [6]. Given a pair of sequences, both methods determine the best alignment, and the calculation time is inversely related to the length of the two sequences [7].

In this paper, we will concentrate on finding a method to align the DNA pair's sequence that addresses the issue of time consumption when using the Needleman-Wunch algorithm. We will do this by adjusting the severity of the algorithm's penalties in addition to using parallel programming that makes use of GPUs to process data while calculating the substitution matrix.The paper's first half covered the fundamental ideas of marital pairwise sequence alignment and its many forms, while the second section discussed some earlier research on the subject. The third portion discussed the suggested approach for creating the Needleman-Wunch algorithms, and the fourth section provided a comparison with earlier research. The outcomes of our suggested methodology were presented in the paper's last part.

2. DNA sequence alignment

DNA is the genetic material that enables the transmission of genetic information from one generation to the next. As organisms evolve through time, their DNA diverges from that of their ancestors. As a result, one of the most important criteria in biological research is the analysis of DNA sequences of living creatures [8]. DNA sequence alignment is a technique for determining the precise nucleotide order in a nucleic acid molecule [9]. It is commonly employed in bioinformatics to ascertain the molecular sequence of an unknown DNA sequence that has been conserved via natural selection throughout evolution [10].

A major problem in bioinformatics is the need for sequence alignment to find biological molecules that are similar, such as proteins, DNA, or RNA. It promotes the formation of numerous biological molecular connections. Sequence alignment may be categorized into two main groups [11]. Pair-wise sequence alignment (PSA), which finds the maximum degree of similarity between two biological sequences, enables one to evaluate the degree of similarities and the possibility of homology. In order to ascertain the evolutionary link between the query sequences, a technique known as multiple sequence alignment (MSA) focuses on aligning three or more biological sequences [12].

DNA is the genetic material that enables the transmission of genetic information from one generation to the next. As organisms evolve through time, their DNA diverges from that of their ancestors. As a result, one of the most important criteria in biological research is the analysis of DNA sequences of living creatures [8]. DNA sequence alignment is a technique for determining the precise nucleotide order in a nucleic acid molecule [9]. It is commonly employed in bioinformatics to ascertain the molecular sequence of an unknown DNA sequence that has been conserved via natural selection throughout evolution [10].

A major problem in bioinformatics is the need for sequence alignment to find biological molecules that are similar, such as proteins, DNA, or RNA. It promotes the formation of numerous biological molecular connections. Sequence alignment may be categorized into two main groups [11]. Pair-wise sequence alignment (PSA), which finds the maximum degree of similarity between two biological sequences, enables one to evaluate the degree of similarities and the possibility of homology. In order to ascertain the evolutionary link between the query sequences, a technique known as multiple sequence alignment (MSA) focuses on aligning three or more biological sequences [12].

2.1 Pairwise sequence alignment

Finding the best piecewise local or global alignments of DNA (nucleic acid) or protein (amino acid) sequences is the goal of pairwise sequence alignment algorithms. The goal of sequence alignment is to place as many similar residues as possible in the same locations [13]. When two sequences are aligned globally, all of the characters in both sequences take part in the alignment. The main use of global alignments is the discovery of closely related sequences [14]. Local sequence alignment is a method of aligning sequence areas based on the density of matches [15].

The identification of identical residue sequences or patterns of identical residues that exist in the sequences in the same order is part of the computational process to compare two or more sequences, and these groups are depicted by writing sequences as follows:

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

In order to ensure that columns include identical symbols from the involved sequences wherever feasible, the sequences are filled with gaps.

Received: January 25, 2023. Revised: February 18, 2023.

2.2 Multiple sequence alignment

Multiple sequence alignment is primarily concerned with identifying biological relationships between various sequences, such as nucleotide or amino acid sequences, in order to research their underlying fundamental properties or activities. In other words, the matched sequences may have greater biological links if they have more similarities between them [16]. We can also anticipate the functionalities or properties of aligned sequences as follows:

```
-  C  G  C  T  -
G  C  G  -  T  -
-  C  -  C  G  T
```

Due to the complexity and inflexibility of directly processing the sequences with their biologically relevant length, computational tools are utilized to construct and analyse the MSAs.

3. Needleman-Wunsch algorithm

The algorithm that uses the idea of dynamic programming for sequence alignment is the Needleman-Wunsch algorithm. Phosphates, sugars, and nitrogenous bases are the three components that make up DNA, as is common knowledge. The DNA's nitrogenous bases, which are represented as the nucleotides adenine, cytosine, guanine, and thymine, constitute its information-containing region. Therefore, in bioinformatics, these nitrogenous bases were represented as A, C, G, and T [17].

The length of nitrogenous bases can occasionally range up to 18000 characters, making it impractical to establish relationships in DNA, proteins, and amino acids using observable correlation. As a result, the Needleman-Wunsch method has been frequently employed to do so. Numerous dynamic programming algorithms exist that provide results that are globally optimal (e.g., Needleman-Wunsch, Smith-Waterman). However, these techniques have a lengthy calculation time [18]. It is divided into two stages [19]:

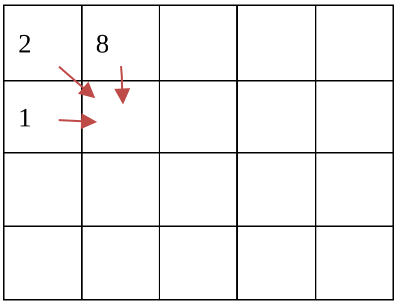


Figure. 1 Needleman-Wunsch substitution matrix

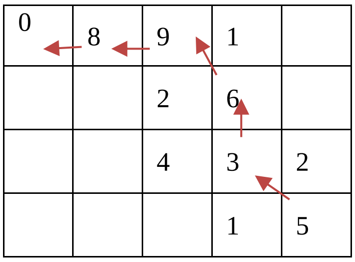


Figure. 2 Backtracking phase in Needleman-Wunsch algorithm

1- Stage 1 Given input sequences x and y of lengths K and L, respectively, calculate the similarity score. In order to implement the Needleman-Wunsch method, a dynamic programming (DP) matrix F is kept up-to-date. The best score for aligning the first characters in x with the first j characters in y may be found in the (i, j) entry, the Eq. (1) is used to initialize and calculate the matrix:

$$R[i,j] = MAX\begin{cases} R[i-1,j-1] + \delta(\text{xi},\text{yi}) \\ R[i-1,j] + \delta(\text{xi},-) \\ R[i,j-1] + \delta(-,\text{yi}) \end{cases} \quad (1)$$

Each cell in the alignment matrix is dependent on its left, upper, and higher-left (diagonal) neighbours, as seen in Fig. 1.

2- Stage 2: Find the ideal backtracking phase of the procedure begins at the final place of the matrix $R_{k,l}$after the matrix has been generated and saved in quadratic storage space. The method returns to the starting cell $R_{0,0}$ after selecting the cell whose elements were used during the prior phase of the process at each step. Every upward movement represents an insertion into sequence S1, and every leftward movement represents an insertion into sequence x or a deletion in sequence y. Either a match or a mismatch would be indicated by a diagonal movement as shown in figure 2:

Algorithm 1 illustrates the Needleman-Wunsch pseudocode used in the alignment procedure:

Algorithm. 1 Needleman-Wunsch algorithm

```
K- the length of sequence 1
L- the length of sequence 2
R-K x Ldimensional dynamic substitution matrix
$\delta(x_i,-)$-score from aligning $x_i$ with a gap
$\delta(-,y_j)$-score from aligning $y_j$ with a gap
$\delta(x_i,y_j)$-score from aligning xi with $y_j$
    1. R[0,0]=0
    2. For i=1 to K Do:
        • R[i,0]=r[i-1,0]+ $\delta$(xi,-)
```

```
3. For j=1 to L Do:
   • $R[0,j]=R[0,j-1]+ \delta(-,y_j)$
   • For i=1 to L Do:
   • $R[i,j] = MAX\begin{cases} R[i-1,j-1] + \delta(\mathrm{xi},\mathrm{yi}) \\ R[i-1,j] + \delta(\mathrm{xi},-) \\ R[i,j-1] + \delta(-,\mathrm{yi}) \end{cases}$
4. Return R[K,L]
```

Output: *Score of Optimal alignment for aligning K and L*
Time Complexity: *O(KL)*

4. Literature survey

In this part, we will present previous studies in order to comprehend the previous methods that is used in DNA pairwise sequence alignment.

A unique sequence alignment method that uses a genetic algorithm (GA) to choose the optimal alignment score between two sequences—which might be DNA or protein sequences—was presented by Gautam Garai and Biswanath Chowdhury in 2015 [20]. By breaking up a large space into smaller subspaces, the recommended genetic-based method, known as cascaded pairwise alignment with genetic algorithm (CPAGA), reduces the search space required. The sequence pair is broken up into several pieces before the alignment procedure starts. This type of reduction improves the search process's ability to find the best global or nearly global solution, even for longer sequences.

BulkAligner, an approach to a graph-based in-memory trinity decentralized system, was put out by Junsu Lee et al. in 2015 [21]. They partition each trimmed sequence into k-mer sequence fragments in order to get the data in graph form. They split the reference sequence into k subsamples (where k = the lengths of 134 sequence fragments). They transform the reference data sets into a graph representation data format akin to the de Bruijn graph in order to create a reference graph for every slave.

In 2016, Sanchita Saha Ray et al. [22] presented a pair-wise DNA sequence alignment method that is memory-efficient. To facilitate the quicker and more accurate identification of the ideal alignment, a novel concept of a pointing matrix is applied. The recommended technique finds the ideal global alignment by employing a dynamic programming paradigm. For lengthy DNA sequences, the method takes significantly less space than the Needleman-Wunsch method. On a random sample of 100 fake samples, which created DNA sequence pairings in 10 distinct circumstances, the complete technique was evaluated. The approach takes 34–42% less time to discover the optimal alignment than another method, although it took a little longer (9–11%) to create.

In 2018 saw the development of two strategies by Sara Q. Abedulridha and Eman S. Al-Shamery [23] to address the pairwise sequence alignment issue. The goal of the first strategy is to divide the DNA tape into portions that match and don't match by splitting it into pieces and using adaptive interleaving windows. A multi-zone genetic algorithm (MZGA) is proposed as an enhanced method in the second strategy. On the basis of cut-points and gap location, a novel crossing approach is proposed. A dataset of DNA with lengths ranging from 66 to 26037 bases was used to evaluate the approach. The proposed method produced the best alignment score for the DNA sequences.

Amr Ezz El-Din Rashed et al. [24] in 2021 utilized two widely used DNA sequence matching techniques using an effective software and hardware digital fusion. The recommended approach focuses on parallelizing each of these fundamental algorithms while adhering to predetermined limitations. Under certain limitations, it relies on the well-known alignment and parallelization algorithms for DNA sequences. In comparison to the present state-of-the-art technology, the recommended MATLAB solution for local and global alignment yields significantly superior elapsed time and GCUPS.

A cache-efficient parallel method to solve the sequence alignment and gap penalty problems for shared-memory devices was presented by Shubham et al. in 2022 [25]. This article describes a successful divide-and-conquer strategy. Higher parallelism and data localization are asymptotically present. It may be possible to increase cache complexity for some dynamic programming problems by using a matrix splitting technique rather than a simple 2-way split of the dynamic programming table across all aspects.

5. Methodology

In order to reduce the Needleman-Wunsch

		A	C	A	A	C	T	G	G	G	T
	0										
A											
C											
C											
A											
A											
C											
T											
G											
A											

Figure. 3 Substitution matrix initialization

Received: January 25, 2023. Revised: February 18, 2023.

algorithm's execution time, a technique to align the pairwise sequence is described in this paper. It involves:

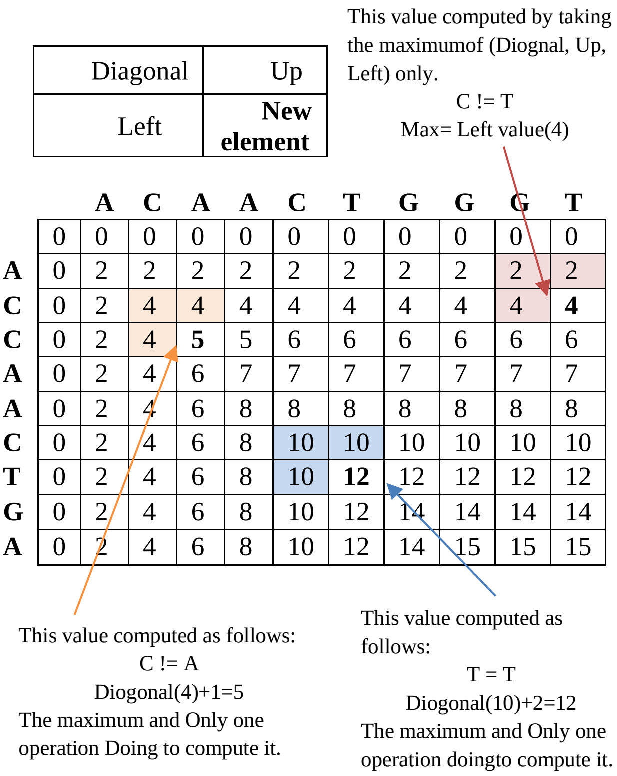
1- Initialization of substitution matrix
2- Managing the score in substitution matrix (match, mismatch, and gap penalty).
3- Creation of two threads to compute the substitution matrix in parallel using GPU.
4- Determining the optimal alignment

5.1 Managing substitution matrix parameters

A tabular box known as a score matrix is used to keep track of score outcomes. The Needleman-Wunsch algorithm's score matrix goes through an initialization phase and computes cell boxes in the end. It includes the following steps:

1. Initialization: To align the original matrix gap, a sequence matrix with M + 1 columns and N + 1 rows is made, where M is the length of the reference sequence and N is the length of query sequence. Letters from the reference sequence fill in the horizontal axis of the matrix that was formed, and similarly, characters from the query sequence filled in the vertical axis and the intersection of the first row and the first column of the initialization matrix is given the value P(0, 0) to 0 prior to the scoring starting from the top left corner to the bottom right corner of the matrix (i.e., the initial gap) as shown in Fig. 3.
2. Determine the match, mismatch, and gap penalty scores: The score for matches and mismatches, as well as the penalty value for gaps, are decided at this stage. It is now possible to select penalties that reduce the Needleman-Wunsch algorithm's computing operations. When using the Needleman and Winch algorithm, four computing processes are carried out to determine the value of each cell in the substitution matrix. These operations are:

   a. The value of the cell in the diagonal, to which a matching or mismatching score is added (according to the matching of the rule in the reference sequence and the query sequence).
   b. The value of the cell at the top is added to the value of the gap penalty.
   c. The value of the cell at the left is added to the value of the gap penalty.
   d. Find the maximum value of the above three steps to be added to the specified cell.



Diagonal	Up
Left	**New element**

This value computed by taking the maximumof (Diognal, Up, Left) only.
C != T
Max= Left value(4)

		A	C	A	A	C	T	G	G	G	T
	0	0	0	0	0	0	0	0	0	0	0
A	0	2	2	2	2	2	2	2	2	2	2
C	0	2	4	4	4	4	4	4	4	4	**4**
C	0	2	4	**5**	5	6	6	6	6	6	6
A	0	2	4	6	7	7	7	7	7	7	7
A	0	2	4	6	8	8	8	8	8	8	8
C	0	2	4	6	8	10	10	10	10	10	10
T	0	2	4	6	8	10	**12**	12	12	12	12
G	0	2	4	6	8	10	12	14	14	14	14
A	0	2	4	6	8	10	12	14	15	15	15

This value computed as follows:
C != A
Diogonal(4)+1=5
The maximum and Only one operation Doing to compute it.

This value computed as follows:
T = T
Diogonal(10)+2=12
The maximum and Only one operation doingto compute it.

Figure. 4 Computation of the value of each cell in the substitution matrix

In our proposed method the maximum value among the three variables was chosen first (the diagonal, the top, and the left). The processes b and c specified in paragraph 2 will not be performed if the maximum value corresponds to the value in the diagonal. Because the gap value is set to zero, and sum time the three arithmetic operations (a, b, and c) will be cancelled when the maximum value represents the upper value or left value so this will speed up the substitution matrix calculation as shown in Fig. 4.

3. Compute the substitution matrix in parallel: The score matrix filling step of the Needleman-Wunsch method is most the time-consuming and parallelizable step. The presence of dependencies between matrix elements each element relies on the top, left, and element diagonally between them presents a considerable barrier to parallelization. This means that threads that traverse the whole matrix cannot be initiated. To make sure that each thread calculates an element only after the elements on which it depends have previously been calculated, a rigidly specified order of operation must be observed. As a result, the following technique is employed:

   a. Threads number: It is known in advance how many threads the program will execute. They are assumed to be two.

Received: January 25, 2023. Revised: February 18, 2023.

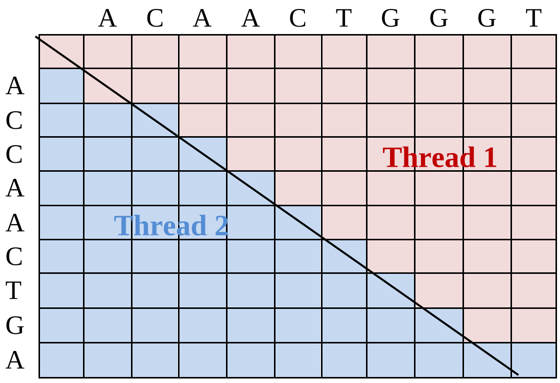


Figure. 5 Thread in substitution matrix

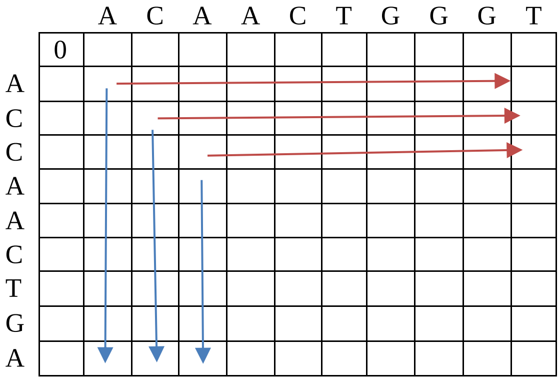
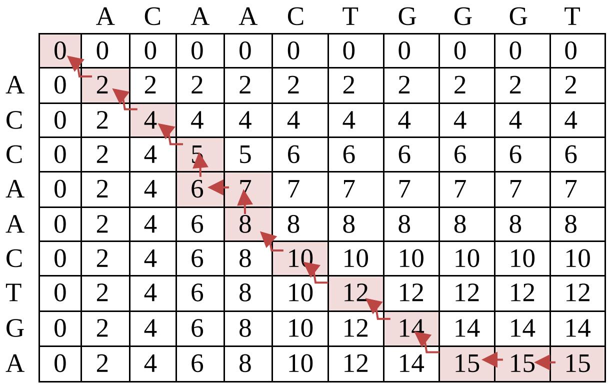


Figure. 6 Parallel computation of substitution matrix cell



		A	C	A	A	C	T	G	G	G	T
	0	0	0	0	0	0	0	0	0	0	0
A	0	2	2	2	2	2	2	2	2	2	2
C	0	2	4	4	4	4	4	4	4	4	4
C	0	2	4	5	5	6	6	6	6	6	6
A	0	2	4	6	7	7	7	7	7	7	7
A	0	2	4	6	8	8	8	8	8	8	8
C	0	2	4	6	8	10	10	10	10	10	10
T	0	2	4	6	8	10	12	12	12	12	12
G	0	2	4	6	8	10	12	14	14	14	14
A	0	2	4	6	8	10	12	14	15	15	15

Figure. 7 Tracing back to determine optimal alignment path

b. Initializations: The matrix's initializations are carried out concurrently since its members are independent of one another as shown in Fig. 5.

c. Matrix division: The matrix is divided into two sections by symmetry, and each section has a parallel thread processing each element at once. In the substitution matrix, one of the two threads moves horizontally (along the rows), and the other moves vertically (along the columns), as shown in Fig. 6.

4. Determining the optimal alignment: The deduction of the optimal alignment from the substitution matrix is done by traceback. The traceback path has three potential moves:

a. Diagonal: two sequences of letters are aligned.
b. Left: A gap is added to the left sequence.
c. Up: A gap is added to the top sequence.

Figure 7 depicts a tracing back operation to determine the optimal path of the reference and query sequencing.

The optimal alignment will be as follows:

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

5.2 Algorithms for proposed method

The primary algorithm for generating the substitution matrix is listed below, along with the fundamental procedures to determine each matrix element:

```
Algorithm. 2 Main algorithm of proposed method
R- Reference Sequence
Q- Query Sequence
M- Reference Length
N- Query Length
S-dynamic substitution matrix M x N
1.$\delta(x_i,-)=0$        //score from aligning $x_i$ with a gap
2.$\delta(-,y_j)=0$//score from aligning $y_j$ with a gap

3.$\delta(xi,yj) = \begin{cases} score\ from\ aligning\ xi\ with\ yj\ = 2 \\ score\ from\ aligning\ xi\ with\ yj\ = 1 \end{cases}$   ...(2)

4.S[0,0]=0
5.Do Thread1 and Thread2 in parallel//Horizontal
  Thread &Vertical Thread
6.Backtracking
7.Return R[M,N]
Output: Score of Optimal alignment for aligning M and N
```

We will now explain the steps we used to develop each thread independently. The first thread divides the replacement matrix diagonally by working on the rows and progressively decreasing in each row. While the second thread operates similarly to the first, but on columns instead of rows:

The actions that are carried out by the horizontal thread to determine the first half of the substitution matrix are included in Algorithm 3.

```
Algorithm. 3 Algorithm of horizontal thread
R- Reference Sequence
Q- Query Sequence
M- Reference Length
N- Query Length
```

```
 S-dynamic substitution matrix M x N

 1.K= -1
 2.For i=1 to M Do:
   • K=K+1
   • For j=1 to N-K Do:
     ○ If R[i-1] = Q[j+k-1] Do
       • S[i,j+k]=S[i-1,j+k-1]+δ(x_i,y_j) //δ(x_i,y_j)=2
         if R[i-1] = Q[j+k-1]

//δ(x_i,y_j)=1
if R[i-1] = Q[j+k-1]
     ○ Else

       • $Maximum = Max\begin{cases} S[i-1,j+k-1] \\ S[i-1,j+k] \\ S[i,j+k-1] \end{cases}$

         • if S[i-1,j+k-1] >=Maximum Do
           ▪ S[i,[j+k] = Maximum +1
         • Else
           ▪ S[i,j+k] = Maximum// Because
               δ(x_i,y_j)=0

 3. Return R[M,N]
```

Output: Score of Optimal alignment for aligning M and N

The actions carried out by the vertical thread to determine the first partition of the substitution matrix is included in Algorithm 4.

Algorithm. 4 Algorithm of vertical thread

```
R- Reference Sequence
 Q- Query Sequence
 M- Reference Length
 N- Query Length
 S-dynamic substitution matrix M x N dimension
 1.K= -1
 2.For i=1 to N Do:
   • K=K+1
   • For j=1 to M-K Do:
     ○ If R[i+k-1] = Q[j-1] Do
         • S[i+k,j] = S[i+k-1,j-1]+ δ(x_i,y_j) //
           δ(x_i,y_j)=2
         if R[i+k-1] = Q[j-1]
// δ(x_i,y_j)=1 if
R[i+k-1] = Q[j-1]
     ○ Else

       • $MAX = \begin{cases} S[i+k-1,j-1] \\ S[i+k-1,j] \\ S[i+k,j-1]) \end{cases}$

         • if S[i+k-1,j-1]>=Max Do
           ▪ S[i+k,j] = Max+1

         • Else
           ▪ S[i+k,j] = Max // Because δ(x_i,y_j)=0

 3. Return R[M,N]
```

Output: Score of Optimal alignment for aligning M and N

The phase of backtracking follows effectively finishing the substitution matrix computation in its entirety in order to extract the best alignment from the substitution matrix after computing all of the values of it. The bottom right cell, which was the last to be completed with the score, serves as the starting point for the traceback and the traceback value recorded in the cell determines how one advances. There are three alternative directions to move: up, left, or diagonally (toward the top-left corner of the matrix). In order to determine the whole path, which is the optimal path, this procedure is continued until the point (0,0) is reached (the best alignment).

6. Results

In order to evaluate the suggested pairwise sequence alignment approach, the suggested parallel dynamic programming-based sequence alignment algorithm approach is simulated on 10 data sets from the national centre for biotechnology information (NCBI), available at https://www.ncbi.nlm.nih.gov. The test was carried out on an 11th Gen Intel(R) Core (TM) i7-11800H @ 2.30GHz system. Each dataset consists of two non-identical sequences, one representing the reference sequence and the other representing the query sequence. The Needleman-Wunsch algorithm was first enhanced by adopting score 2 for the nucleotides that match in the two sequences (Reference and Query) and score 1 for the mismatched nucleotides, in addition to the algorithm's most crucial step, which is the adoption of score 0 as a penalty for the gap. It helped reduce the algorithm's arithmetic operations. By comparing the execution time of the proposed method with the execution time of the Needleman and Wunsch algorithm, it was found that the algorithm made a considerable contribution to reducing the processing time. Table 1 shows the difference in time when executing the 10 datasets of different lengths:

The execution time of Needleman-Wunsch algorithm is minimized between 21% and 23%, according to the first assessment of 10 datasets. After

Table 1. Comparison between the suggested method and Needleman-Wunsch in time consuming

Dataset	Reference Length	Query Length	Time consumed in NW algorithm(second)	First improvement in Time consumed in proposed method (second)	Time comparison ratio
1	19070	19051	440.161	336.880	23%
2	17883	17867	376.459	293.863	22%
3	5705	5700	38.198	29.897	22%
4	2502	2502	7.354	5.742	22%
5	1820	1820	3.951	3.031	23%
6	13327	13510	213.275	168.358	21%
7	3564	3564	15.340	11.810	23%
8	1572	1572	2.910	2.265	22%
9	8669	8669	89.277	69.656	22%
10	1478	1465	2.51140	1.951	22%

Table 2. Comparison between the improvement in suggested method and Needleman-Wunsch in time consuming

Dataset	Reference Length	Query Length	Time consumed in NW algorithm (second)	Second improvement in Time consumed in proposed method (second)	Time comparison ratio
1	19070	19051	440.161	45.862	90%
2	17883	17867	376.4591	41.408	89%
3	5705	5700	38.198	4.981	87%
4	2502	2502	7.354	1.817	75%
5	1820	1820	3.951	1.449	63%
6	13327	13510	213.275	24.982	88%
7	3564	3564	15.340	2.633	83%
8	1572	1572	2.910	1.371	53%
9	8669	8669	89.277	10.094	89%
10	1478	1465	2.511	1.332	47%

that, we considered how to further decrease the execution time and we discovered that by splitting the substitution matrix into two parts and running each partition of the matrix on a separate thread, one of which works horizontally (on rows) and the other vertically (on columns), in parallel, we could get faster execution times. On the same 10 datasets that

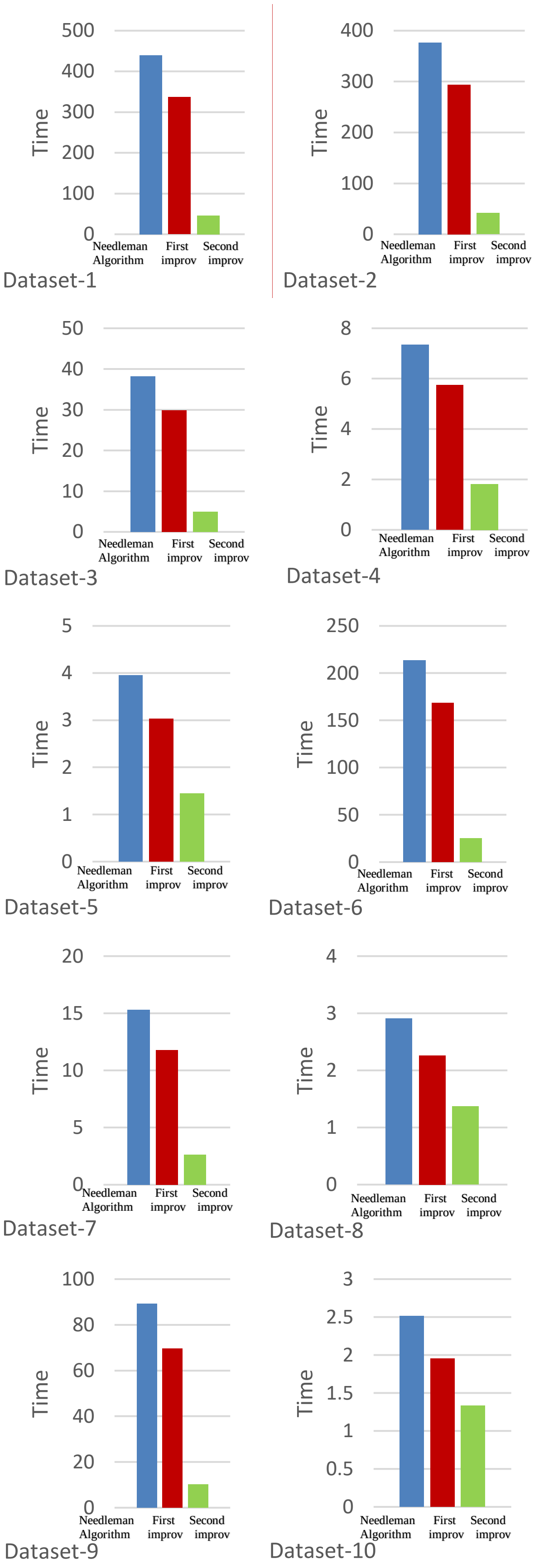


Figure. 8 Comparison between the suggested method and Needleman-Wunsch in time consuming

we utilized in the prior stage, we tested this notion and found the execution time has been improved by processing on the NVIDIA GeForce RTX 3060 Laptop GPU platform with a CORE i7 2.30GHz CPU to enhance the performance of the suggested method. The time difference between using Needleman's method and the first and second algorithmic advancements is displayed in Table 2.

Through the comparison we conducted in Table 2 on the various datasets with lengths, we discovered that the percentage of time improvement for our proposed method grows noticeably as the lengths of the sequences increase, with the percentage ranging between 47% and 90% as shown in Fig. 8 which illustrates the advancement over time.

7. Comparative study

The approaches from previous studies that were examined in section 4 of this research have been compared to the suggested technique. Following is a summary of the comparison:

The "local minimum" problem, which results from the greedy nature of the algorithm, is discovered to be the basic drawback of this technique when comparing the suggested model with the model described in reference [20, 23]. Because further sequences are added to the alignment, any errors produced in any intermediate alignments cannot be fixed. Furthermore, it is not possible to determine if one alignment is better than another or whether the best alignment for a given case has been determined using an objective function. In contrast to the model that we presented, which finds the pair alignment in such a way that there are no errors in the final product of the process, the model that we proposed has an optimal solution in the final product.

The study's authors in paper [21] used graph theory and the trinity, which states that graphs employ several indications and might be challenging to manage. It can cause serious memory complications. While the model that we have provided does not cause memory problems during the construction of the substitution matrix that is used to create pairwise alignments, there are still some advantages to using this model.

The authors in [22] developed a pair-wise DNA sequence alignment method that is memory-efficient. Using 100 random fake samples, which created DNA sequence pairings in ten distinct circumstances, the complete technique was evaluated. The optimal alignment is obtained after a prolonged running period, while the primary benefit of the proposed work is that it significantly reduces the amount of time spent to determine the optimal alignment.

Table 3. Comparative analysis of the proposed procedure and the preliminary research

Reference	Maximum DNA Sequence Length	Accuracy
[20]	1771	98.99%
[21]	120	96%
[22]	1024	-
[23]	2157	96%
[24]	4411	98.3%

The researchers described a lookup table-based approach for matching the paired DNA sequences in the reference [24]. By reading this paper, we can observe that the researchers emphasized the necessity for reference and query sequences to be of comparable length. The need for these two sequences limits the applicability of this approach to diverse sequences. On the other hand, the proposed model is capable of finding the alignment of reference and query sequences even when the lengths of the sequences are different from one another.

The divide-and-conquer strategy was employed by the researchers in reference [25] to determine the alignment of DNA sequences. This method's drawback is that the algorithm produces a sub-optimal alignment. In contrast to the model that we put forth, which results in the optimal sequence alignment.

From the aforementioned and by consulting earlier literature that is concerned with determining the pairwise alignment of DNA sequences, it was discovered that the approach proposed by us is capable of providing an optimal sequence alignment in a shorter amount of time and for various lengths of sequences. Table 3 outlines the maximum length that was used for DNA data as well as the accuracy level reached by the suggested methodology in comparison to earlier research.

8. Conclusion

The computerized biological sequence alignment program is facing previously unheard-of difficulties as a result of the exponential expansion of biological sequence data. As the foundation for sequence analysis, sequence alignment limits the application and precision of future analysis. In this study, we offer a novel approach for constructing sequence alignments of DNA sequences that relies on Needleman-Wunsch's technique for computing substitution matrices and controls the penalties for each condition, including match, mismatch, and gap conditions. The Needleman-Wunsch algorithm relies on four fundamental operations to calculate the substitution matrix, which is represented by gathering the value of the score of similarity or dissimilarity, the value in the diagonal of the element to be calculated, and two arithmetic operations that include the process of adding the gap penalty value with the upper and left values, The fourth operation's function is to determine the largest value of the outcomes of the first three operations, which is the value to be computed. This was discovered when the extent of the effect of the penalty values on the speed of the algorithm was first investigated. Here, it became apparent to us that if we first calculate the maximum value of the diagonal, upper, and left values of the value to be calculated, and only then calculate the value of the gap penalty to zero, it will allow us to reduce the mathematical operations into two operations (finding the maximum value, which represents the value of the diagonal, and adding it to the score of congruence or mismatch, which is 2 in the case of a match and 1 in the case of a mismatch), the procedures are sometimes reduce to just one (finding the maximum value)where the maximum value corresponds to one of the two (upper or left) of the value to be determined. On 10 datasets from the NCBI website, we applied the suggested procedure, and we discovered that there is a time difference between (21% and 23%), this was Needleman's algorithm's initial improvement. Following that, we considered how to advance our technology, and we came to the conclusion that it is possible to divide the substitution matrix into two halves diagonally, with each half's values being calculated by a single thread and in parallel, depending on the development in the devices, including the performance and capabilities of the GPU, which helped to reduce the computation time for the dataset Ten by (47% and 90%) by performing multiple operations at the same time. We compared the proposed approach to earlier studies and discovered that, in contrast to the majority of these methods, which mandate equal lengths for both sequences, our approach is able to deal with the alignment of pairwise sequences with different lengths of DNA sequences (reference and query) in less time. From here, it becomes apparent that our technique, as opposed to Needleman-Wunsch algorithm and other previously employed approaches, can provide an optimal pairwise sequence alignment in a short amount of time.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Rana H. Saloom provided the idea, technique, software, formal analysis, materials, data collection, and writing-original version preparation. Hussein K. Khafaji provided supervision, revision, and editing.

References

[1] J. Parihar, P. Kansal, K. Singh, and H. Dhiman, “Assessment of bioinformatics and healthcare informatics”, In: *Proc. of 2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 465–467, 2019.

[2] S. E. Metwally, O. M. Ouda, and M. Helmy, *Next Generation Sequencing Technologies and Challenges in Sequence Assembly*, Vol. 7. Springer Science & Business, 2014.

[3] S. Makigaki and T. Ishida, “Sequence alignment using machine learning for accurate template-based protein structure prediction”, *Bioinformatics*, Vol. 36, No. 1, pp. 104–111, 2020.

[4] E. J. Moyer and A. Das, “Motif identification using cnn-based pairwise subsequence alignment score prediction”, *ArXiv Preprint arXiv:2101.08385*, 2021.

[5] H. Khaled, R. E. Gohary, N. L. Badr, and H. M. Faheem, “Accelerating pairwise DNA Sequence Alignment using the CUDA compatible GPU”, *Int J ComputAppl*, Vol. 84, No. 1, 2013.

[6] Y. S. Lee, Y. S. Kim, and R. L. Uy, “Serial and parallel implementation of Needleman-Wunsch algorithm”, *International Journal of Advances in Intelligent Informatics*, Vol. 6, No. 1, pp. 97–108, 2020.

[7] Y. Jararweh, M. A. Ayyoub, M. Fakirah, L. Alawneh, and B. B. Gupta, “Improving the performance of the needleman-wunsch algorithm using parallelization and vectorization techniques”, *Multimed Tools Appl*, Vol. 78, No. 4, pp. 3961–3977, 2019.

[8] H. Kaur and L. Chand, “Pairwise sequence alignment of biological database using soft computing approach”, In: *Proc. of 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 72–77, 2016.

[9] R. H. Saloom and H. K. Khafaji, “A Survey for the Methods of Detection and Classification of Genetic Mutations”, *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 28, pp. 1796–1816, Dec. 2022.

[10] B. Chowdhury and G. Garai, “A review on multiple sequence alignment from the perspective of genetic algorithm”, *Genomics*, Vol. 109, No. 5–6, pp. 419–431, 2017.

[11] H. C. Ng, S. Liu, and W. Luk, “Reconfigurable

acceleration of genetic sequence alignment: A survey of two decades of efforts", In: *Proc. of 2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–8, 2017.

[12] T. Lassmann, *Kalign 3: multiple sequence alignment of large datasets*, Oxford University Press, 2020.

[13] V. Gancheva and I. Georgiev, "Multithreaded parallel sequence alignment based on needleman-wunsch algorithm", In: *Proc. of 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 165–169, 2019.

[14] C. Hong and A. H. Tewfik, "Designing a tighter searching space for pairwise global sequence alignments over multiple scoring systems", In: *Proc. of 2007 15th European Signal Processing Conference*, pp. 733–737, 2007.

[15] Y. J. Song, D. J. Ji, H. Seo, G. B. Han, and D. H. Cho, "Pairwise heuristic sequence alignment algorithm based on deep reinforcement learning", *IEEE Open J Eng Med Biol*, Vol. 2, pp. 36–43, 2021.

[16] M. A. Nayeem, M. S. Bayzid, A. H. Rahman, R. Shahriyar, and M. S. Rahman, "Multiobjective formulation of multiple sequence alignment for phylogeny inference", *IEEE Trans Cybern*, 2020.

[17] P. Malathi, M. Manoaj, R. Manoj, V. Raghavan, and R. E. Vinodhini, "Highly improved DNA based steganography", *Procedia Comput Sci*, Vol. 115, pp. 651–659, 2017.

[18] C. Kyal, R. Kumar, and A. Zamal, "Performance-Based Analogising of Needleman Wunsch Algorithm to Align DNA Sequences Using GPU and FPGA", In: *Proc. of 2020 IEEE 17th India Council International Conference (INDICON)*, pp. 1–5, 2020.

[19] X. Xu, Y. Chan, K. Xu, J. Zhang, X. Wang, Z. Yin, and W. Liu, "SLPal: Accelerating long sequence alignment on many-core and multi-core architectures", In: *Proc. of 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2242–2249, 2020.

[20] G. Garai and B. Chowdhury, "A cascaded pairwise biomolecular sequence alignment technique using evolutionary algorithm", *Inf Sci (N Y)*, Vol. 297, pp. 118–139, 2015.

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

[22] S. S. Ray, A. Banerjee, A. Datta, and S. Ghosh, "A memory efficient DNA sequence alignment technique using pointing matrix", In: *Proc. of 2016 IEEE Region 10 Conference (TENCON)*, pp. 3559–3562, 2016.

[23] S. Q. Abedulridha and E. S. A. Shamery, "Optimal Pair DNA Sequence Alignment based on Matching Regions and Multi-Zone Genetic Algorithm", *International Journal of Engineering & Technology*, Vol. 7, No. 4.19, pp. 751–756, 2018.

[24] A. E. E. D. Rashed, M. Obaya, H. El, and D. Moustafa, "Accelerating DNA pairwise sequence alignment using FPGA and a customized convolutional neural network", *Computers & Electrical Engineering*, Vol. 92, p. 107112, 2021.

[25] S. Prakash and P. Ganapathi, "An Algorithm for the Sequence Alignment with Gap Penalty Problem using Multiway Divide-and-Conquer and Matrix Transposition", *Inf Process Lett*, Vol. 173, p. 106166, 2022.