Fast Dynamic Algorithm for Sequence Alignment based on Bioinformatics

Sara A.Shehab
Faculty Of Computer& Information
Menofya University

Arabi Keshk
Faculty Of Computer& Information
Menofya University

Hany Mahgoub
Faculty Of Computer& Information
Menofya University

ABSTRACT
Sequence alignment is widely used in Bioinformatics for Genome Sequence difference identification. It is the main problem of computational biology. Any sequence of Deoxyribonucleic acid (DNA), Ribonucleic acid (RNA), and protein can be alignment by many algorithms called bioinformatics algorithms. This paper presents a new implemented algorithm for sequence alignment based on concepts from bioinformatics algorithms .The implemented algorithm is called fast dynamic algorithm for sequence alignment (FDASA). This implemented algorithm based on making a matrix of M×N (M is the length of the first sequence, N is the length of the second sequence), After that filling the three main diagonal without filling the unused data and at the same time get the optimal solution; so that the execution time is decreased, the performance is high and the memory location decreased. The implementation introduced in this paper made a comparison between the dynamic algorithms Needleman-Wunsch algorithm, Smith-Waterman and our algorithm FDASA to test the execution time. The results show that our algorithm FDASA decreased the execution time when compared with Needleman-Wunsch and Smith-Waterman algorithms.

General Terms
Bioinformatics.

Keywords
computational biology; FDASA; Sequence; DNA; RNA; dynamic algorithms; Needleman-Wunsch; Smith-Waterman.

1. INTRODUCTION
In bioinformatics, the main problem of the computational biology, text processing and pattern recognition is string matching. Many algorithms have been designed to overcome this problem to make a comparison between two strings or evaluate the longest common subsequence that two string share [1].There are two types of Bioinformatics algorithms:-(a)-dynamic algorithms such as Needleman-Wunsch and Smith-Waterman algorithms, these algorithms has advantages and disadvantages. The advantage :- finds the optimal alignment solution between the sequences, the disadvantage:- takes more time to make the alignment this decrease the performance. (b)-the heuristic algorithms such as BLAST and FASTA, these algorithms also has advantages and disadvantages. The advantage :- ignores the unused data from computation this speed the performance,

the disadvantage:- the alignment isn't the optimal solution. The new algorithm based on achieving the advantages of both dynamic algorithms and the heuristic algorithms.

The length of normal DNA sequence range from 40KB to 60 KB in FASTA format [17]. It take a long time to aligning two DNA sequences [15] [3] [4] [5] on a single processor. The complexity of aligning two sequences is O (M×N) where M is the length of first sequence and N is the length of the second sequence. our Algorithm is globally align [6][7] two DNA Sequences, it reduces the time to O(3M+1)if the 2 sequence have same length or O(3M+2)if the 2 sequence have different length. We have implemented a new algorithm for handling the DNA matching and alignment problem. There are to approaches to make the algorithms:-

1.1 Global Alignment
Global alignment get the maximum match between the sequences as it assume that the two sequences are similar. This alignment attempts to match the two sequences from the end to the end even though if they are different in some parts [9] [2].

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

1.2 Local Alignment
Local alignment searches for the part of the two sequences that match well. The output of this alignment is the region that have a big similarity, according to some criterion are considered. Using the same sequences as above, one could get [9][2]:

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

Most commonly used algorithm for local alignment is Smith-Waterman algorithm [10].global alignment is better as it get a maximum match between the sequences. Our algorithm globally alignment.

2. RELATED WORKS
There are many Pair-Wise algorithms used to compare two sequence. Fig. 1 illustrates an alignment between the sequences A=.ACAAGACAGCGT and B=.AGAACAAGGCGT [14]

```
A = ACAAGACA G -  CGT
    |  ||  |  ||   |||
B = AGAACA – AG G CGT
```

Fig 1: Alignment of two sequences.

There are several pervious works in this field of DNA sequence alignment, these works based on the Bioinformatics dynamic algorithms like Needleman-Wunsch algorithm and Smith-Waterman, first the Needleman-Wunsch outlined in details as follow:-

2.1 Needleman-Wunsch Algorithm

In the Needleman-Wunsch algorithm [16], this algorithm first take the two sequences and create a 2-dimensional array with the length of Multiply of the two sequence's length, each cell can be evaluated from the maximum of the three cells around it and at the same time keep a pointer of the maximum value to make the trace-back to get optimal solution [11][12][13].

Fig. 2 and Fig. 3 shows the matrix filled with values and pointers.

		0	1	2	3	4	5	6	7	8	9	10	11	12
		-	A	G	A	A	C	A	A	G	G	C	G	T
0	-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12
1	A	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
2	C	-2	0	0	-1	-2	-1	-2	-3	-4	-5	-6	-7	-8
3	A	-3	-1	-1	1	0	-1	0	-1	-2	-3	-4	-5	-6
4	A	-4	-2	-2	0	2	1	0	1	0	-1	-2	-3	-4
5	G	-5	-3	-1	-1	1	1	0	0	2	1	0	-1	-2
6	A	-6	-4	-2	0	0	0	2	1	1	1	0	-1	0
7	C	-7	-5	-3	-1	-1	1	1	1	0	0	2	1	0
8	A	-8	-6	-4	-2	0	0	2	2	1	0	1	1	2
9	G	-9	-7	-5	-3	-1	-1	1	1	3	2	1	2	1
10	C	-10	-8	-6	-4	-2	0	0	0	2	2	3	2	1
11	G	-11	-9	-7	-5	-3	-1	-1	-1	1	3	2	4	3
12	T	-12	-10	-8	-6	-4	-2	0	0	0	2	2	3	5

Fig 2: Filled Needleman-Wunsch Matrix & Trace-back

F(i,j)		i=0	1	2	3	4
			A	G	T	A
j=0		0	-1	-2	-3	-4
1	A	-1	1	0	-1	-2
2	T	-2	0	0	1	0
3	A	-3	-1	-1	0	2

Fig 3: Filled Needleman-Wunsch Matrix &Trace-back

Every non-decreasing path from (0, 0) to (M,N) corresponds to a global alignment of the two sequences.

The steps of the Needleman-Wunsch algorithm is as follows[2]:-

Initialization

$F(0, 0) = 0$

$F(0, i) = -i * d$

$F(j, 0) = -j * d$

Main Iteration

For each i = 1 . . . M

For each j = 1 . . . N

$F(i, j) = \max$

$\{ F(i-1, j-1) + s(xi, yj)$, case 1

$F(i-1, j) - d$, case2

$F(i, j-1) - d$, case3 $\}$

$Ptr(i, j) =$

{ DIAG , if case 1

LEFT , if case 2

UP , if case 3 }

Termination

F(M,N) is the optimal score, and from Ptr(M,N), we can trace back the optimal alignment. the optimal alignment for fig. 2 is in the fig. 1, the optimal alignment for fig. 3 is as follow(fig.4):-

A G T A

A - T A

Fig 4: Alignment of two sequences.

Performance

Time: O(NxM) (We need to fill out the whole matrix)

Space: O(NxM) (We need a matrix to store all the trace back pointers).

After the Needleman-Wunsch algorithm discussed in details, the problem of these dynamic algorithms is that it take more time to fill all the matrix of the two sequences, although there are unused data but it must be found to help in filling all the cells in the matrix, so these algorithm take many times to make this computation, Tahir Naveed try to decrease the execution time by using the Parallel Needleman-Wunsch algorithm[2]. A parallel version of Needleman-Wunsch algorithm [2] has been developed; which uses multiple processors for initializing, Calculating and filling the DataMatrix (Stores the DNA Sequences and their calculated values) and the PointerMatrix (Stores DNA Sequences and the pointer values to be used later in backtracking). Tahir Naveed algorithm doesn't include backtracking process to keep record for values to be calculated in each iteration on parallel machines. This algorithm has been implemented on Grid using Alchemi Framework [8]. All the matrices in parallel version of Needleman-Wunsch algorithm are places in global memory space so that all available processors can access them at the same time to perform initialization and other calculations. By developing Needleman's parallel algorithm we have Tahir Naveed reduced the calculation time from O(NxM) to O(N+M) by using two CPU to make a parallel computation[2].

2.2 Smith-Waterman

In 1981, T. F. Smith and M. S. Waterman [18],[14] showed that alignment between two sequence can be computed locally with the same idea as Needleman and Wunsch. The main difference is that F[i, j] add to the maximum function that declared in Needleman and Wunsch the possibility of Zero value. The formula for computing F[i, j] becomes:

$F[i, j] = \max \{ 0;$

$F[i-1, j-1] + sub (A[i], B[j]);$

$F[i-1, j] + del (A[i]);$

$F[i, j-1] + ins (B[j]) \}$

3. PROPOSED ALGORITHM

The previous work in the Bioinformatics algorithms depend on making a matrix of M×N (M is the length of the first sequence, N is the length of the second sequence) and filling all the cells in this matrix, this filling operation takes more execution time, decrease the performance and increase the number of the memory location used to make this sequence comparison. The new implemented algorithm FDASA decrease the execution time, fast the performance and decrease number of the memory location used to make this sequence comparison. The main steps in the new algorithm FDASA shown in the following flowchart (see fig. 9).

The algorithm steps is as following:-

Given two sequences A and B. Create a matrix M×N (where M is the length of first sequence, N the length of second sequence). Every non-decreasing path from (0, 0) to (M, N) corresponds to a global alignment of the two sequences, there are five main steps for the FDASA algorithms as follow:

1. Initialization
Gap= -1
Match= +1
Mismatch= -1
C (0, 0) = 0
C (0, 1) = C (0, 0) + Gap
C (j, 0) = C (0, 0) + Gap

2. Detect the three main diagonals in the matrix.

3. Main Iteration

For each cell in the three main diagonals:
For each i = 1 . . . M
For each j = 1 . . . N
if (i = j)
C (i, j) = (C (i − 1, j − 1) + match)
Dir (i, j) = (DIAG)
if (i ≠ j)
C (i, j) = max
{C (i − 1, j − 1) + mismatch, case 1
C (i − 1, j) + gap, case2
C (i, j − 1) + gap, case3}
Hint: the three values in the maximum function needn't be found it must be at least one value.
dir (i, j) =
{DIAG, if case 1
LEFT, if case 2
UP, if case 3}

4. Termination
C (M, N) is the optimal score, and from dir (M, N), we can trace back the optimal alignment.

5. Performance
Time: O (3M + 1) if the 2 sequence have same length or O (3M + 2) if the 2 sequence have different length.

Space: O (3M + 1) if the 2 sequence have same length or O (3M + 2) if the 2 sequence have different length.

3.1 FDASA Case Study (1) :-

This case study assume that the length of the two sequences are differ, so the memory space can be calculated from the formula O(3M+2),and the execution time O(3M+2). Given Sequence A="AGTA"and sequence ="ATA"
Create a matrix of size M*N (M is the length of first sequence; N is the length of second sequence).
The main steps of FDASA algorithm is as follow:-

1. Initialization
Gap=-1
Match=+1
Mismatch=-1
dir= the direction of the trace back for optimal alignment.
C(0, 0) = 0
C(0, 1) = C(0,0)+Gap
= -1
F(j, 0) = C(0,0) +Gap = -1

2. Detect the three main diagonals in the matrix.

3. Main Iteration
Calculate the values for each cell in the three main diagonals:
C(1,1)=C(0,0)+match=0+1=1 (i=j)
dir=diagonal
C(1,2)= max(C(0,1),C(1,1),C(0,2)) +mismatch (i≠j)
C(0,2) not have any value so
C(1,2)=max(C(0,1),C(1,1)) +mismatch
= max(-1,1)+(-1) =1-1=0
dir= the maximum value =left
And so on for all the 3 diagonal values, the matrix will be as follow (fig.5):

	-	**A**	**G**	**T**	**A**
-	0	-1			
A	-1	1	0		
T		0	0	+1	
A			-1	0	2

Fig.5 Fill 3 diagonal values and trace back pointer

4. Termination

C(M,N) is the optimal score . The Optimal Score is 2. trace back the optimal alignment from this optimal score position by the dir value: so the optimal alignment is as follow(fig.6):-

A G T A
| | |
A - T A

Fig 6: Alignment of two sequences.

The Score is:-=(Number of Match)*match value +(Number of mismatch)*mismatch value+(number of gapped values)*gap value
=(3)*1+(0)*-1+(1)*-1=3-1=2

5. Performance
Time: O(3M+1) if the 2 sequence have same length or O(3M+2)if the 2 sequence have different length.
= O(3*M+2) = O(11)
Space: O(3M+1) if the 2 sequence have same length or O(3M+2)if the 2 sequence have different length.
= O(3*M+2) = O(11)

3.2 FDASA Case Study (2) :-

This case study assume that the length of the two sequences are same, so the memory space can be calculated from the formula O(3M+1),
and the execution time O(3M+1).

Given Sequence A="ACAAGACAGCGT" and sequence B="AGAACAAGGCGT"
Create a matrix of size M*N . The main steps of FDASA algorithm is as follow:-

1. Initialization

Gap=-1
Match=+1
Mismatch=-1
dir= the direction of the trace back for optimal alignment.
C(0, 0) = 0

C(0, 1) = C(0,0)+Gap
= -1
F(j, 0) = C(0,0) +Gap
= -1

2. Detect the three main diagonals in the matrix.
3. Main Iteration

Calculate the values for each cell in the three main diagonals:
C(1,1)=C(0,0)+match=0+1=1 (i=j)
dir=diagonal
C(1,2)= max(C(0,1),C(1,1),C(0,2)) +mismatch (i≠j)
C(0,2) not have any value so
C(1,2)=max(C(0,1),C(1,1)) +mismatch
= max(-1,1)+(-1)=1-1=0
dir= the maximum value =left
And so on for all the 3 diagonal values, the matrix will be as follow (fig 7):

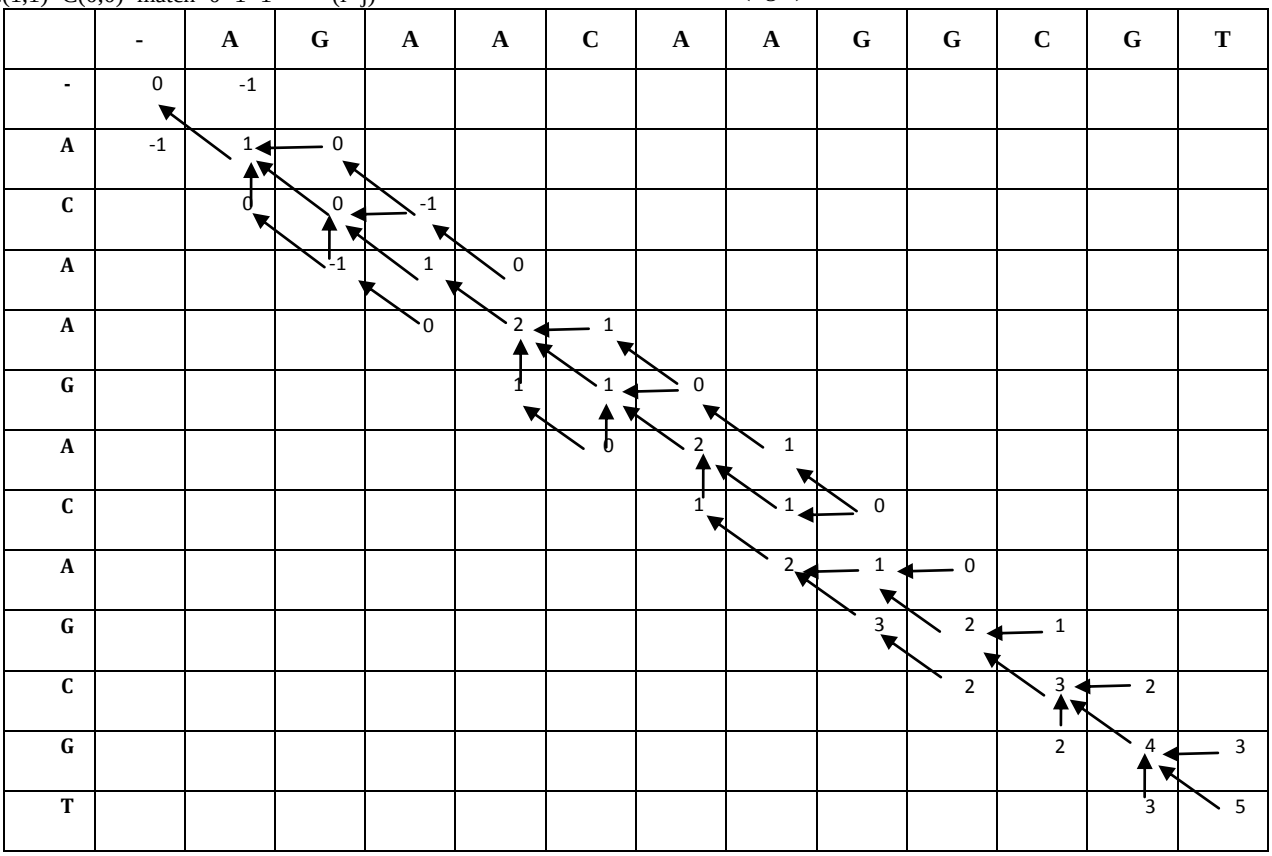

Fig.7 Fill 3 diagonal values and trace back pointer

4. Termination

C(M,N) is the optimal score .The Optimal Score is 5 trace back the optimal alignment from this optimal score position by the dir value, so the optimal alignment is as follow(fig.8):-

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

Fig 8: Alignment of two sequences.

The Score is:-=(9)*1+(2)*-1+(2)*-1=9-2-2=5

5. Performance

Time: = O(3*M+1) = O(37)
Space: = O(3*12+1) = O(37)

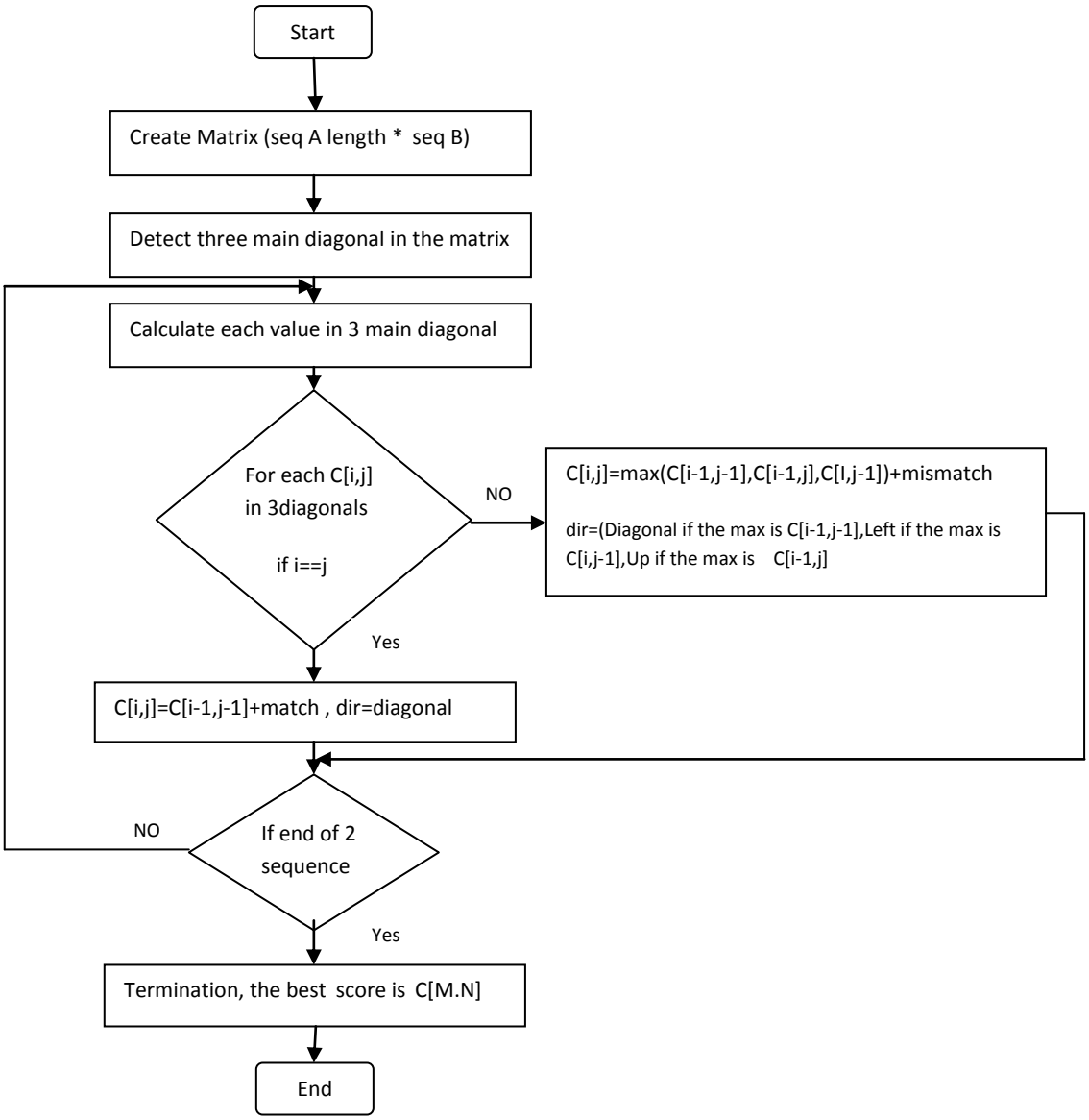

Fig 9: the flow chart of the FDASA algorithm

4. EXPERMINTAL RESULTS

In the implementation we use the old implemented algorithms Needleman-Wunsch algorithm, Smith-Waterman and our new algorithm FDASA to make comparison between them to test the execution time, in fig. 10 the GUI for the algorithms[14]. in fig.11 the output optimal alignment and the score for the sequences A=.ACAAGACAGCGT and B=.AGAACAAGGCGT. and fig. 12 show another case for the new algorithm FDASA to find optimal solution for the sequences A=.ATA. and B=.AGTA. From the fig. 1 and fig.11 and also fig. 4 and fig.12 we found that the FDASA algorithm achieve the same optimal solution as the old algorithms and the same scores , but the execution time are different; the FDASA algorithm decrease the execution time .

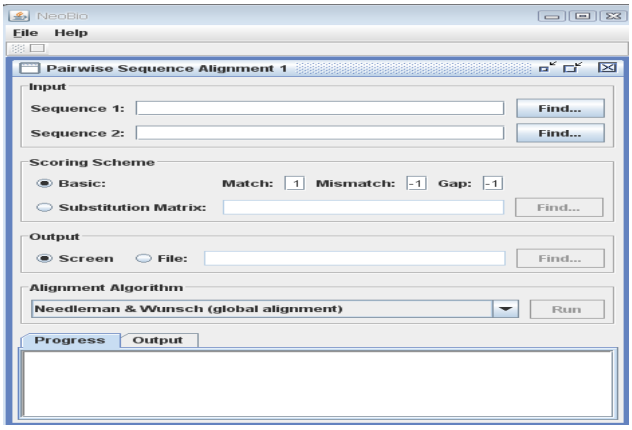

Fig 10: the GUI for all algorithms[3]

Table 1 show Our Algorithm Evaluation Against The Needleman-Wunsch Algorithm And Smith-Waterman we found that in first case when the sequences is

A=.ACAAGACAGCGT and B=.AGAACAAGGCGT the total execution time for the alignment by using Needleman-Wunsch algorithm is approximately ~=1.273millisecond, and approximately ~=1.042millisecond by the Smith-Waterman algorithm , and finally approximately ~=0.163millisecond by FDASA algorithm. We also found that in second case when the sequences is A=.ATA. and B=.AGTA. the total execution time for the alignment by using Needleman-Wunsch algorithm is approximately ~=0.140 millisecond, and approximately ~=0.139 millisecond by the Smith-Waterman algorithm , and finally approximately ~=0.053 millisecond by FDASA algorithm. From these values we found that our algorithm FDASA achieve the least execution time this come from ignoring the unused data of the matrix and evaluate the only three main diagonal. Fig. 5 and fig.7 show the output matrix of the FDASA algorithm with the trace-back pointer when computing the three main diagonal and ignoring the unused data according to FDASA algorithm.

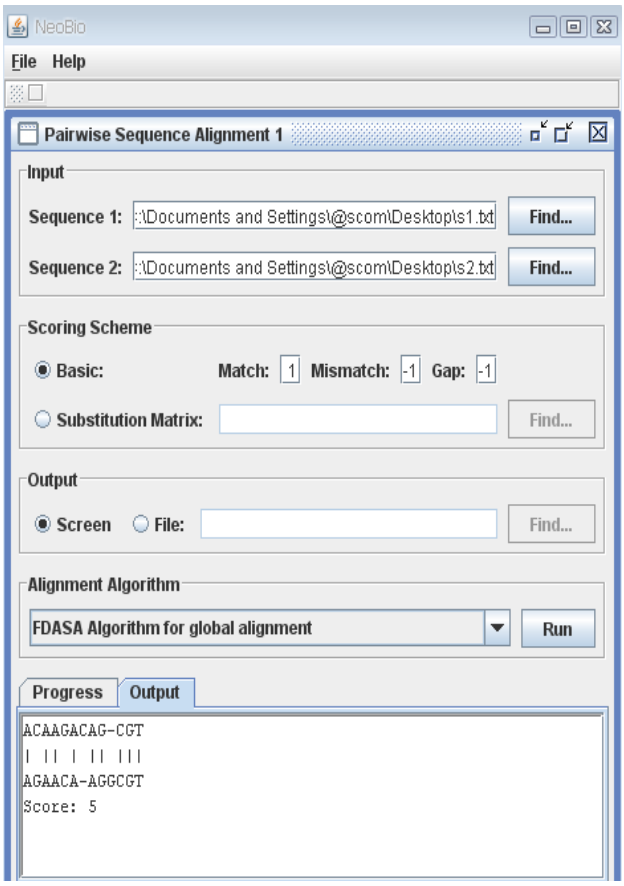

Fig 11: the output alignment for the sequences A=.ACAAGACAGCGT. and B=.AGAACAAGGCGT. by using the new algorithm FDASA

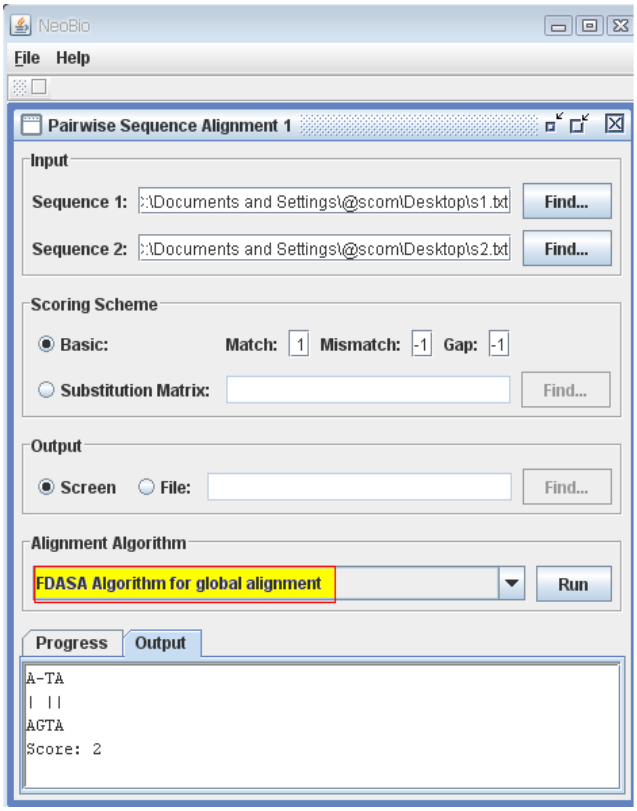

Fig 12: the output alignment for the sequences A=.ATA. and B=.AGTA. by using the new algorithm FDASA.

When comparing fig.5 with fig. 3 we found the three main diagonal is the same and the trace back pointer to calculate the optimal solution is the same ,as shown:-

A G T A
A - T A

and the optimal score is the same =2.

Also fig.7 when compared with fig. 2 we found the three main diagonal is the same and the trace back pointer to calculate the optimal solution is the same ,as shown:-

A = ACAAGACAG-CGT

| || | || |||

B = AGAACA-AGGCGT

and the optimal score is the same =5. From these results , FDASA algorithm find the same optimal solution as the Needleman-Wunsch algorithm, Smith-Waterman however, it ignore most data in the matrix , and also the FDASA algorithm reduce the execution time for the alignment and also speed the performance and reduce the memory location used for this comparisons.

Table 1. Our Algorithm Evaluation Against The Needleman-Wunsch Algorithm And Smith-Waterman algorithms

Algorithms	Execution Time(fig. 11)	Execution Time(fig.12)	Performance	Memory Locations	Big O Notation
FDASA	~0.163 Millisecond	~0.053 Millisecond	High	O(3M+1)if two sequence have the same length O(3M+2)if the sequence have different length	O(3M+1)if two sequence have the same length O(3M+2)if the sequence have different length
Needleman-Wunsch algorithm	~1.273 Millisecond	~0.140 Millisecond	Low	O(M*N)as we need to fill all the matrix	O(M*N)as we need to fill all the matrix
Smith-Waterman	~1.042 Millisecond	~0.139 Millisecond	Low	O(M*N)as we need to fill all the matrix	O(M*N)as we need to fill all the matrix

5. CONCLUTION

This paper presented a new implemented algorithm for sequence alignment based on bioinformatics algorithms. This implemented algorithm overcomes some problems of bioinformatics algorithms by ignoring the unused data. By implementing the FDASA algorithm we have reduced the time. The main idea of the new implemented algorithm is reduce the execution time, speed the performance and decrease the memory location used to make the sequence comparisons. This implemented algorithm based on taking the advantage of dynamic algorithms that is get the optimal solution for the sequences alignment, and also take the advantage of the heuristic algorithm that it is decrease the execution time for the sequence comparisons. In the implementation we use java language and the Net-beans 6.5 IDE with the JDK 1.6 to test the algorithms under the Windows Operating system with RAM 1G.

6. ACKNOWLEDGMENTS

I would like to thank my Mom, My Husband, all my family, and All Doctors who help me in this research.

7. REFERENCES

[1] Dimitris Papamichail and Georgios Papamichail2, “Improved algorithms for approximate string matching (extended abstract)”,BMC Bioinformatics 2009

[2] Tahir Naveed, Imitaz Saeed Siddiqui, Shaftab Ahmed, “Parallel Needleman-Wunsch Algorithm for Grid”, Proceedings of the PAK-US International Symposium on High Capacity Optical Networks and Enabling Technologies (HONET 2005), Islamabad, Pakistan, Dec 19 - 21, 2005.

[3] The Science Behind the Human Genome Project (online), Oak Ridge National Laboratory. Available: http://www.ornl.gov/sci/techresources/Human_Genome/project/info.shtml

[4] Facts About Genome Sequencing (online), Oak Ridge National Laboratory. Available: http://www.ornl.gov/sci/techresources/Human_Genome/faq/seqfacts.shtml

[5] Source of DNA Sequences (online), National Center for BiotechnologyInformation.Available: http://www.ncbi.nlm.nih.gov/mapview

[6] Pairwise Sequence Comparison (online), Lab of Bioinformatics,Institute of Computing Technology (ICT), Chinese Academia of Sciences(CAS). Available: http://www.bioinfo.org.cn/lectures/index-13.html

[7] Introduction to Bioinformatics - Chapter 5 - Introductory SequenceAnalysis (online), Human Genome Mapping ProjectResource Centre(HGMP-RC) by UK Medical Research Council. Available:http://portal.rfcgr.mrc.ac.uk/Courses/Jemboss_3day/Chapter5.html#Global%20sequence%20alignment

[8] Krishna N. and Akshay L. and Dr. Rajkumar B., 2002, Alchemi v0.6.1 Documentation (online), University of Melbourne. Available: http://alchemi.net/

[9] BioInformatics Educational Resources Documentation (online), European Bioinformatics Institute United Kingdom. Available: http://www.ebi.ac.uk/2can/tutorials/protein/align.html

[10] ChittaBaral, Computational Molecular Biology, CSE 591 Arizona State University, United States of America. Available:http://www.public.asu.edu/~cbaral/cse591-03/classnotes/seq-align.pdf

[11] Rong X, Jan 2003, Pairwise Alignment - CS262 - Lecture 1 Notes(online), Stanford University. Available: http://ai.stanford.edu/~serafim/cs262/Spring2003/Notes/1.pdf

[12] Bin Wang, 2002, Implementation of a dynamic programming algorithm for DNA Sequence alignment on the Cell Matrix Architecture (online), Utah State

University, Logan, Utah. Available: http://www.cellmatrix.com/entryway/products/pub/wang 2002.pdf

[13] Chand T. John, April 2004, CS273: Algorithms for Structure andMotion in Biology, Stanford University. Available:http://www.stanford.edu/class/cs273/scribing/8 .pdf

[14] SérgioAnibal de Carvalho Junior ,"Sequence Alignment Algorithms", thesis, 2002/2003.

[15] About the Human Genome Project (online), Oak Ridge NationalLaboratory.Available: http://www.ornl.gov/sci/techresources/Human_Genome/ project/about.shtml

[16] Needleman S, Wunsch.,"A general method applicable to the search for similarities in the amino acid sequences of two proteins", J Mol Biol. 1970, 48:443-453.

[17] FASTA Format Description (online), NGFN-BLAST. Available:http://ngfnblast.gbf.de/docs/fasta.html

[18] Smith, T. F. and M. S. Waterman, Identification of common molecular subsequences,Journal of Molecula Biology, 147:195-197, 1981.