International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 8– No.1, December 2014 – www.ijais.org

A Noble Approach on Bioinformatics: Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR)

Paramita Basak Upama, Jarin Tasnim Khan, Zeba Yasmin, Farah Zemim, Nazmus Sakib
Department of Computer Science & Engineering
Ahsanullah University of Science and Technology, Dhaka, Bangladesh

ABSTRACT
Alignment generally means lining up characters of strings, allowing matches and mismatches and also the characters of one string to be placed opposite spaces made in opposing strings. Sequence alignment arises in many fields, such as: molecular biology, inexact text matching (e.g. spell checkers; web page search), speech recognition and many more. This Paper presents a new implemented algorithm for sequence alignment based on the concepts of some already established algorithms of Bioinformatics which is named as "Smart Sequence Alignment Algorithm applying DNA Replication", in short "SSAADR". It aligns two sequences of any length, based on filling up a matrix of size m X n where m is the length of the first sequence and n is the length of the second sequence, and using the techniques of both global and local alignment. To make the proposed algorithm faster, the theory of DNA replication has also been introduced here, according to which, while using the alternative sequences of both DNA sequences instead of the original ones, the execution time of the proposed algorithm SSAADR decreases immensely.

Keywords
Sequence alignment, DNA, RNA, Dynamic Algorithms, Z Algorithm, Boyer-Moore Algorithm, Needleman-Wunsch (NW) Algorithm and Smith-Waterman (SW) Algorithm.

1. INTRODUCTION
Sequence alignment is a robust filed which can intelligently retrieve information of all the existence of presence from different patterns on various disciplines like biological, image processing, phrases and pattern matching. Sequences alignment is significant to find out the associative information regarding nature, formation and behavior of the patterns. In this paper, the concept of sequence alignment of bioinformatics is used [1] [2]. It is the way of arranging the sequences of Deoxyribonucleic Acid (DNA), Ribonucleic Acid (RNA) and Protein [3] [4] [5]. There are two algorithms of bioinformatics which are mostly used: Needleman-Wunsch Algorithm and Smith-Waterman Algorithm [1] [6].They are dynamic algorithms used to find optimal alignment solution between two strings [7] [8] [9] [10] [11].

The NW algorithm aligns two strings by applying global alignment technique, where the SW algorithm performs the same job by implementing local alignment technique. The NW algorithm runs faster than the SW algorithm, though the second one gives the best scores for alignment. But both of them take pretty long time for their executions, which are really a big problem in practical places where long strings are to be aligned.

The newly proposed Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm of this paper adopts the concepts of both NW and SW algorithms and uses their best parts in its own implementation. In this algorithm, the optimal alignment solution is found in much less time than the previously discussed ones. In Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm the concept of DNA replication is also used. While the alternative strings of the original reference and test sequences are used for executing the algorithm, the result is found faster than using the original sequences.

2. LITERATURE REVIEW

2.1 DNA Replication:
DNA is duplicated before a cell divides. The reason for this duplication is two strands of a DNA molecule which have complementary base pairs. The nucleotide sequence of each strand automatically supplies the information needed to produce its partner. Each strand can be used as a template to produce a complementary strand when the two strands of DNA molecule are separated. After that one template and one new complement together form a new DNA double helix [12] [13].

2.2 Global & Local Alignment:
Global alignment finds the maximum matches between two sequences assuming that the two sequences are similar. It attempts to match both these sequences from the end to the end even if they are different in some parts [2] [14] [15].On the other hand, local alignment searches for the part of two sequences that match well. Here, the sequences are segmented in arbitrary length. This alignment finds the region in these sequences that have a big similarity like a substring, according to some criterion [2] [15] [16] [17].

2.3 Z Algorithm:
The Z algorithm gathers the information needed for speeding up the string matching process where each element of the string is matched with its prefix [18] [19]. In the algorithm, $\underline{Z_i}$ is the length of the longest substring of S that starts at i and matches a prefix of S. Here, Z-box is the box at i starts at i and ends at $i+Z_i-1$. However, r_i is the right-most endpoint of the Z-boxes that begin at or before I and l_i is the left endpoint of the Z-box ends at r_i.

2.4 Boyer-Moore Algorithm:
This algorithm only preprocesses the string being searched for (the pattern), but not the string being searched in (the text) using the information gathered during the preprocess step to skip sections of the text [20]. A shifting can be calculated by applying two rules. The maximum one of them

is an actual shifting offset. The rules are: i) bad character rule that is basically right shifts the test string to get a match with the mismatched bad character and ii) good suffix rule which performs right shifts the test string to get match with the mismatched character without disturbing the already aligned suffix of it.

2.5 Needleman-Wunsch Algorithm:

The basic idea of Needleman-Wunsch algorithm is to build up the best alignment by using optimal alignments of smaller subsequences. The Needleman-Wunsch algorithm requires a 2-D array or matrix of score and also requires alignment score for a pair of residues to be >=0. Gap penalty is not required. Score cannot be decreased between two cells of a pathway [14] [21].

There are three basic steps in this algorithm:

a) Initialization of the Matrix
b) Matrix fill-up

Three different values can be obtained from that take the maximum among them and fill the i^{th} and j^{th} position with the score obtained. Each position $M_{i,i}$ is defined to be the maximum score at position i,j

$M_{i,i}$ = MAXIMUM **[**
$M_{i-1, i-1} + S_{i,,i}$ (match or mismatch in the diagonal)
$M_{i, i-1} + w$ (gap in sequence #1)
$M_{i-1, j} + w$ (gap in sequence #2)**]**

c) Trace back to align the sequences

Let, ACTGATTCA as sequence 1, and ACGCATCA as sequence 2. The length of the sequence 1 and sequence 2 are 9 and 8 respectively. The initial matrix is created with m+1 column's and n+1 row's (where m and n corresponds to length of the sequences). Extra row and column is given, so as to align with gap, at the starting of the matrix. After trace backing the possible sequence alignment will be:

For Seq#1: **A C – G C A T – C A**
| | | | | | |
For Seq#2 :**A C T G – A T T C A**

2.6 Smith-Waterman Algorithm:

The Smith-Waterman Algorithm determines on finding an optimal alignment by considering any location, length and sequence of alignments. Scores or weights are assigned to each character-to-character comparison based on calculations. For exact matching and substitutions, scores are positive while negative for insertions and deletions. The highest scoring alignment is reported by adding the scores in score matrices. There are only three changes:

a) Instead of increasing gap penalties, the edges of matrix are initialize to 0.

b) The maximum score is never less than 0, and no pointer is recorded unless the score is greater than 0.

c) The trace-back starts from the highest score in the matrix (rather than at the end of the matrix) and

ends at a score of 0 that it first encounters (rather than the start of the matrix).

The only difference in Smith-Waterman algorithm from Needleman-Wunsch algorithm is:

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]) }

Let, CGTGAATTCAT as sequence 1 or A and GACTTAC as sequence 2 or B. The two sequences are arranged in a matrix form with A+1columns and B+1rows. The values in the first row and first column are set to zero. In this example we can see the maximum score in the matrix as 18, which is found in two positions that lead to multiple alignments, so the best alignment has to be found.

G A A T T C A G A A T T – C
| | | | | | | | | | | | |
G A C T T – A G A C T T A C

+ + - + + - + + + - + + - +
5 5 3 5 5 4 5 5 5 3 5 5 4 5

3. PROPOSED SMART SEQUENCE ALIGNMENT ALGORITHM APPLYING DNA REPLICATION (SSAADR)

The previously discussed algorithms in this paper works by filling a matrix of size m x n (m is the length of the first sequence, n is the length of the second sequence).The new implemented Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm does both global and local alignment using the concepts of Needleman-Wunsch (NW) and Smith-Waterman algorithms (SW).In SSAADR algorithm, the concept of the NW is used for the trace-back and SW is used for main iteration. Initialization of first row and first column of matrix M (i , j) is done with zeros. Trace back starts from the last cell (M (m x n)) and ends at the first cell (M (0,0)).As Smith-Waterman performs faster than Needleman-Wunsch, the concept of SW is used for filling the matrix. For trace back purpose, the concept of NW is used.

To make the proposed algorithm faster, the theory of DNA replication has been introduced here. While the alternative sequences of both original DNA sequences(i.e. sequences found after putting A in place of T and C in place of G and vice versa) are used instead of the original ones, the execution time of the proposed algorithm SSAADR decrease immensely. This conclusion came to light after experimenting with Original-Original, Original-Alternative, Alternative-Original and Alternative-Alternative pairs of DNA sequences and the optimal execution time was found for Alternative-Alternative pair. The whole proposed algorithm is presented here in four segments.

a)*Definitions:*

GP: Gap penalty
M: Highest score
S: Match/mismatch score

Alternative sequence: Sequence formed after placing A in place of T and G in place of C and vice versa in the original sequence

b)*Character arrays:*

S1 [100]: The fixed string (size=a);
S2 [100]: The test string (size=b)

AlignedS1 [100]: Aligned fixed string (size=g);
AlignedS2 [100]: Aligned test string (size=h)

Integer arrays:

```
arr [100][100] :The scoring matrix
SS [100][100] :The substitution matrix
```

Initialization:

```
GP=-1, S=-1, M=-1,
arr [0][0]=0 ,
SS [0][0]=-1;

For each j=1…….b {
                    arr [0][j]=0 ,SS [0][j]=-1
                    } ;
For each i=1…… a {
                    arr [i][0]=0, SS [i][0]=-1
                    };

For each i=0……a-1 Get alternative of S1 in S1;
For each i=0…….....b-1 Get alternative of S2 in S2;
```

c)*Main Iteration:*

```
For each i=1…a {
                    For each j=1…b {

                    if (element of S1=element of S2)
                    then S=1;else S=-1;

          arr [i][j]=max {

                    arr [i-1][j-1] + S;
                    arr [i-1][j]+ GP;
                    arr [i][j-1]+ GP;

                    0}

          M=max{
                    M;arr[i][j]

                    }

                    SS[i][j]=S ;
                              }
                    }
e=a, f=b, k=0;
```

d)*Trace back:*

```
Do {
          Trace back starting with e=e…1 and
          f=f…1 from the last cell of the matrix;

          Increase k by 1 in each iteration;
 }while ( ( e>0 )  & ( f>0 ) );

If (e=0 & f>0) {
                    For each x=f…0 { AlignedS1 [k] = '_' ;
                    AlignedS2  [k] = S2[x-1];  k=k+1;
                                        }
                    }
else if (f=0 & e>0){
                    For each y=e…0
                    {
                    AlignedS1 [k]=S1[y-1];
                    AlignedS2 [k]='_';
                    k=k+1;
                    }
                    }
```

For each i=0…g-1 Get alternative of AlignedS1 in AlignedS1
For each i=0…h-1 Get alternative of AlignedS2 in AlignedS2
The final scoring matrix using SSAADR algorithm for string 1=TAAGCTACC and string 2=TTAAGGACCT is given below:

		T	**T**	**A**	**A**	**G**	**G**	**A**	**C**	**C**	**T**
	0	**0**	0	0	0	0	0	0	0	0	0
T	0	↖ 1	↖ **1**	← 0	0	0	0	0	0	0	↖ 1
A	0	↑ 0	↖ ↑ 0	↖ **2**	↖ ← 1	← 0	0	↖ 1	← 0	0	↑ 0
A	0	0	0	↖ ↑ 1	↖ **3**	← 2	← 1	↖ 1	↖ ← 0	0	0
G	0	0	0	↑ 0	↑2	↖ **4**	↖ ← 3	← 2	← 1	← 0	0
C	0	0	0	0	↑1	↑**3**	↖ 3	↖ ← 2	↖ 3	↖ ← 2	← 1
T	0	↖ 1	↖ 1	← 0	↑0	↑2	↖ ↑**2**	↖ 2	↑1	↖ 2	↖ 3
A	0	↑ 0	↖ ↑ 0	↖ 2	↖ ↑1	↑1	↖ ↑1	↖ **3**	← 2	↑ ← 1	↑ 2
C	0	0	0	↑ 1	↖ 1	↖ ← ↑0	↖ ↑0	↑ 2	↖ **4**	↖ ← 3	← 2
C	0	0	0	↑ 0	↖ ← 0	↖ 0	↖ ← ↑0	↑ 1	↖ ↑3	↖ **5**	← **4**

Figure 1: The final scoring matrix with trace back

4. EXPERIMENTAL RESULTS

The concepts of old implemented algorithms like Needleman-Wunsch algorithm, Smith-Waterman algorithm have been used in the implementation of this proposed Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm. So a comparison has been taken place between these three algorithms (Needleman-Wunsch, Smith-Waterman, Smart Sequence Alignment Algorithm applying DNA Replication) on the basis of their execution time.

In the implementation stage programming language C has been used on the CodeBlocks 13.12 under the Windows 7 Operating System with RAM of 4 GB and Intel Core i3 processor. These values proves that the proposed Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm achieves the least execution time by applying DNA replication theory and by combining the best parts of NW and SW algorithms' techniques.

The following example shows the scores and the final alignments of two test strings (each of which is of 64 bits long) for the algorithms Needleman-Wunsch algorithm, Smith-Waterman algorithm and the proposed Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm. From the example it is evident that the Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm has the lowest time of the three. It has happened because Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm applies matrix fill up style of Smith-Waterman algorithm with trace back style of Needleman-Wunsch algorithm, accompanied by the theory of DNA replication for string sequences.

For experiment 2 strings of 64 character has been taken. Stings are generated randomly.

String 1:
TTTTAAAAGGGGCCCCAAAACCCCTTTTGGGGCCCCT
TTTGGGGAAAAGGGGCCCCTTTTAAAA

String 2:
ATGCCGATCGATCGGAATCGAAGCATTGACGTACCG
ATCAATCGGGACCCAGTTCCGATATTCG

The result of the alignment using Needleman-Wunsch algorithm is Score: -15 and Time: 0.054 sec. The alignment is following, as the string is very long, so it is presented here in several lines:

String 1: **_ T T _ T _ T _ A _ A A _ A _ G G _ _ _ G G _ C**
String 2: **_ T T _ T _ T _ A _ A A _ A _ G G _ _ _ G G _ C**

String 1: **_C _ C C _ A A A _ A C _ C C C T T T T G _ G _**
String 2: **_C _ C C _ A A A _ A C _ C C C T T T T G _ G _**

String 1: **GG_ C _**
String 2: **GG _ C _**

String 1: **C C C _ T _ T T _ _ _ T _ G G G G A _ A _ A _ A**
String 2: **C C C _ T _ T T _ _ _ T _ G G G G A _ A _ A _ A**

String 1: **G_ G _ G G C C _ C _ C T _ T T T _ A _ A A A**
String 2: **G_ G _ G G C C _ C _ C T _ T T T _ A _ A A A**

The result of the alignment using Smith-Waterman algorithm on the same two strings is performed Score: 6 and Time: 0.065 sec. The alignment is following:

String 1: **T T T T _ A A A A _ G _ G G G _ C C C C A**
String 2: **_ _ _ T C _ _ A A T _ C G G G G A _ C C C A**

The result of the alignment using Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm is Score: 6 and Time: 0.045 sec. The alignment is following, as the string is very long of 64 bits, so it is presented here in several lines:
String 1: **_ _ _ _ _ _ _ _ _ _ A A A _ _ A T T _ _ _ T T**
String 2: **_T A C G G C T A G _ _ _ C T A _ _ G C C T T**

String 1: **C C C _ _ C G G G G _ _ _ T _ T _ T T G _ _ _**
String 2: **_ _ _ A G C _ _ _ _ T T C _ G T A _ _ _ A C T**

String 1: **G _**
String 2: **G C**

String 1: **G G _ _ A A A A _ _ _ _ C C C C _ _ _ G G G**
String 2: **_ _ A T _ _ _ _ G G C T _ _ _ _ A G T _ _ _**

String 1: **_ _ G A A A A _ _ _ _ C _ C _ C _ _ C T _ T T**
String 2: **TA G _ _ _ _ C C C T _ G _ G _ G T C _ A _ _**

String 1: **T _ _**
String 2: **_ A G**

String 1: **C C C _ C G _ G G G _ _ A A A A T _ T T T _**
String 2: **_ _ _ G C _ T _ _ _ A T _ _ A A _ G _ _ _ C**

Table 1 shown below is a performance table. It shows the performance of the three algorithms (Needleman-Wunsch algorithm, Smith-Waterman algorithm and the proposed Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm) based on their execution time. In the case where each sequence is 16 bit long which has been taken randomly, let

String 1:
A=TAAGCTACGTAGCAGG

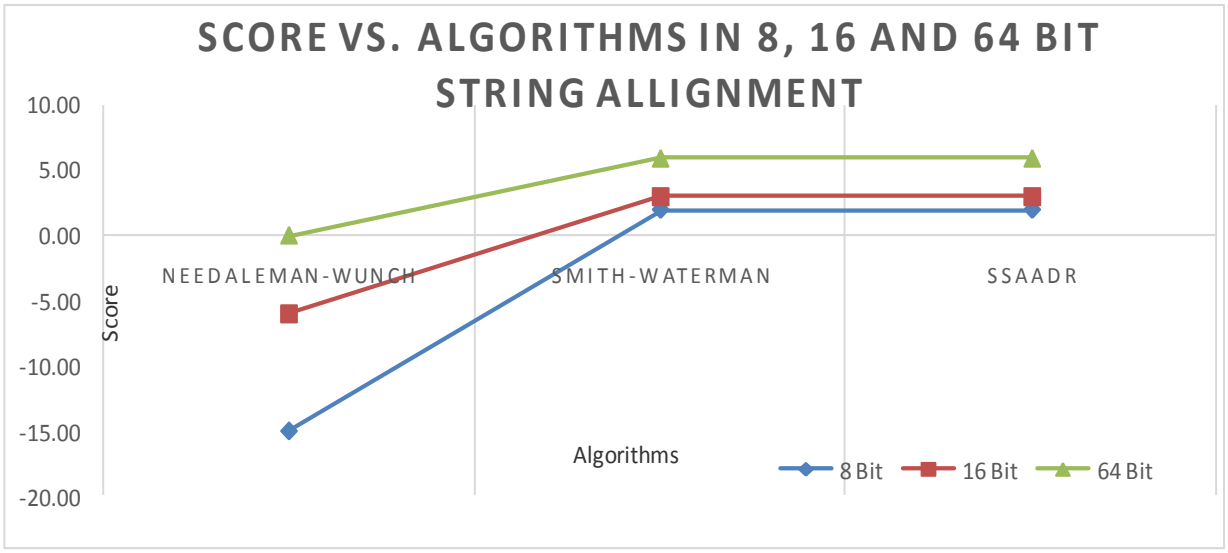
String 2:
B=ATTCGATGGATCGTCT

In the Graph 1, these two 16 bits long strings, 64 bits long strings and some 8 bits long strings which are randomly generated, used for comparison on scores of the Needleman-Wunsch algorithm, Smith-Waterman algorithm and the proposed Smart Sequence Alignment Algorithm applying DNA Replication (SSAADR) algorithm.

The total execution time for aligning the strings are: using Needleman-Wunsch algorithm ~0.0.52second, using Smith-Waterman algorithm ~0.055second and finally ~0.049 second using SSAADR algorithm. In another case where each sequence is 64 bit long. The total execution time for aligning the strings are: using Needleman-Wunsch algorithm ~0.0.54 second, using Smith-Waterman algorithm ~0.065 second and finally ~0.045 second using SSAADR algorithm.

Table 1. Evaluation of the Proposed Algorithm against the Needleman-Wunsch Algorithm and Smith-Waterman Algorithm

Algorithms	Execution Time (for 16 bit)	Execution Time (for 64 bit)	Performance	Big O Notation
SSAADR	~0.049second	~0.045second	High	O(M*N)
Needleman-Wunsch	~0.052second	~0.054second	Medium	O(M*N)
Smith-Waterman	~0.055second	~0.065second	Low	O(M*N)



Graph1:Score Comparison with 8,16 & 64 bit string on differrent discussed algorithm

5. CONCLUSION

This thesis paper presents a new implemented algorithm for sequence alignment based on DNA replication theory. This implemented algorithm overcomes some problems of popular NW and SW algorithms. By implementing the SSAADR algorithm the execution time is reduced and the efficiency is greater than other algorithms. SSAADR algorithm based on taking the advantage of dynamic algorithms, works to get the optimal solution for the sequences alignment, and also take the advantage of the heuristic algorithm that it is to decrease the execution time for the sequence comparisons. The future work of this proposed algorithm will be detecting the genome of a founded fossil by applying codon detection technique in SSAADR algorithm to get the genome of it from the long DNA string & to cross-reference it with the genomes that already saved in a database to get the percentage of match between those two genomes. That result will tell that from which animal or plant kingdom that fossil is from.

6. REFERENCES

[1] N. Islam, Faster and efficient algorithm for sequence alignment, 2012.

[2] M. S. I. Mahmud, M. A. Hosen, Saroer-E-Azam, M. M. Hawlader and Abdullah Al-Mamun, "A Novel Two-Tier Multiple Sequence Alignment Algorithm," in *13th International Conference on Computer and Information Technology (ICCIT 2010)*, 2010.

[3] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody and J. Baldwin, "Initial sequencing and analysis of the human genome," *Nature*, pp. 860-921, 2001.

[4] B. Erickson and P. Sellers, "Recognition of patterns in genetic sequences," pp. 55-91, 1983.

[5] "Exploring Life's Origins: What is RNA?," [Online]. Available: http://exploringorigins.org/rna.html .

[6] V. S. Krishna, P. A. Rasool and D. N. Khare, "String Matching and its Applications in Diversified Fields," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 219-224, 2012.

[7] M. Crochemore, A. Czumaj, L. Gasieniec, S. Jarominek, T. Lecroq, W. Plandowski and W. Rytter, "Speeding Up Two String-Matching Algorithms," *Algorithmica*, vol. 12, no. 4-5, pp. 247-267, 1994.

[8] D. Papamichail and G. Papamichail, "Improved algorithms for approximate string matching," *BMC Bioinformatics*, pp. 1-18, 2009.

[9] S. A. de Carvalho Junior, "Sequence Alignment Algorithms," pp. 4-12, 2003.

[10] D. Knuth, V. Pratt and J. . H. Morris, "Fast pattern matching in strings," *SIAM Journal on Computing*, vol. 6, no. 2, pp. 323-350, 1977.

[11] A. Hume and D. M. Sunday, "Fast string searching," *Software—Practice & Experience*, vol. 21, no. 11, pp. 1221-1248, 1991.

[12] N. Kommajosyula, "Regulation of DNA Replication Origins in Fission Yeast: A Dissertation," *GSBS Dissertations and Theses*, pp. 1-12, 2009.

[13] Watson, J.D. & Crick and F.H.C., "Molecular structure of nucleic acids - A structure for deoxyribose nucleic acid," *Nature*, vol. 171, pp. 737-738, 1953.

[14] S. Needleman and C. Wunsch, "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, 1970.

[15] S. F. Altschul, "Global and Local Sequence Alignment," pp. 2-34.

[16] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195-197, 1981.

[17] S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. . J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403-410, 1990.

[18] I. Yurchenko, "Z algorithm | Ivan Yurchenko," [Online]. Available: http://ivanyu.me/blog/2013/10/15/z-algorithm/.

[19] Codeforces, "Z Algorithm-Codeforces," [Online]. Available: http://codeforces.com/blog/entry/3107.

[20] R. S. Boyer and J. S. Moore, "A Fast String Searching Algorithm," *Communications of the ACM (New York, NY, USA: Association for Computing Machinery)*, vol. 20, no. 0, pp. 762-772, 1977.

[21] Amrita.vlab.co.in, "Global alignment of two sequences - Needleman-Wunsch Algorithm (Theory) : Bioinformatics Virtual Lab II : Biotechnology and Biomedical Engineering : Amrita VishwaVidyapeetham Virtual Lab," [Online]. Available: http://amrita.vlab.co.in/?sub=3&brch=274&sim=1431&cnt=1. .

[22] N. F. A. Saliman, N. D. A. Sabri, S. A. M. A. Junid, N. . M. Tahir and Z. A. Majid, "Smith-Waterman Algorithm Traceback Optimization using Structural Modelling Technique," *International Journal of Simulation,Systems,Science & Technology*, pp. 64-65, 2008.

[23] S. A. Shehab, A. Keshk and H. Mahgoub, "Fast Dynamic Algorithm for Sequence Alignment based on Bioinformatics," *International Journal of Computer Applications (0975 – 8887)*, vol. 37, no. 7, pp. 54-60, 2012.

[24] B. Strengholt and M. Brobbel, "Acceleration of the Smith-Waterman algorithm for DNA sequence alignment using an FPGA," pp. 11-19, 2013.

[25] A. Wozniak, "Using video-oriented instructions to speed up sequence comparison," *Computer applications in the biosciences: CABIOS*, vol. 13, no. 2, pp. 145-150, 1997.