Biological Sequence Matching using Boolean algebra vs. Fuzzy Logic

Nivit Gill
Department of Computer Science and Engineering,
PEC University of Technology,
Chandigarh

Shailendra Singh
Department of Computer Science and Engineering,
PEC University of Technology,
Chandigarh

ABSTRACT

Biological sequence alignment is one of the crucial tasks of computational bioinformatics, and provides base for other tasks of bioinformatics. In this paper, we discuss two different approaches to sequence matching – Boolean algebra and fuzzy logic. First method is a two-valued logic whereas the second is a multi-valued logic. Both the methods perform sequence matching by direct comparison method using the operations of Boolean algebra and fuzzy logic respectively. To ensure the optimal alignment, dynamic programming is employed to align the sequences progressively. Both the methods are implemented and then tested on few sets of real biological sequences taken from NCBI bank and their performances are compared with the CLUSTALW algorithm.

General Terms

Bioinformatics, Sequence Alignment.

Keywords

Sequence alignment, Boolean algebra, Fuzzy Logic, Sequence matching, global alignment, dynamic programming

1. INTRODUCTION

All living organism cells are composed of genetic codes that are passed from one generation to other. This is the reason for some living organisms being biologically similar and some being distinct. The genetic code can be represented as a sequence of alphabets, such as four base pairs of DNA and RNA, or twenty amino acids of protein. These sequences are called biological sequences, and over time, a lot of changes, called mutations, occur in these sequences. The field of bioinformatics aims to align a large number of biological sequences with the purpose of deriving their evolutionary relationships through comparative sequence analysis.

With the help of bio-informatics, computations are applied to the biological sequences in order to analyze and manipulate them. The key idea is to discover and record the role of genetics in an organism's biological characteristics. Sequence alignment is the most basic and essential module of computational bio-informatics and has varied applications in sequence assembly, sequence annotation, structural and functional prediction, evolutionary or phylogeny relationship analysis.

Biological sequence alignment is a field of research that focuses on the development of tools for comparing and finding similar sequences of amino acids or DNA base pairs with the help of computers. The degree of similarity is used to measure gene and protein homology, classify genes and proteins, predict biological function, secondary and tertiary protein structure, detect point mutations, construct evolutionary trees, etc. A sequence alignment refers to the method of arranging biological sequences in order to search similar regions in the sequences. The sequences with high degree of similarity have similar structure and function, and such sequences help in deriving evolutionary or phylogenetic relationships among organisms.

In this paper, we compare two methods of biological sequence matching. The first method employs Boolean algebra, which is a logical calculus of truth values, i.e. 0 or 1, or truth or false. In this method, the given biological sequences are encoded in binary form, and then Boolean operators are applied to determine the percentage of matching of sequences. The second method is based on Fuzzy Logic, which is a form of multi-valued logic derived from fuzzy set theory. The given biological sequences are compared pair wise so as to determine the number of matches, and mismatches between them. Then these counts are fuzzified using fuzzy membership functions, and then fuzzified counts are put in an aggregate fuzzy function in order to find the fuzzy match value of the two sequences. In both the methods, the match value, so calculated, is used to order the sequences according to the similarity. The most similar pair is aligned first and the rest of the sequences are then aligned to this aligned pair.

The outline of this paper is: Section 2 discusses the basics of sequence alignment and its types and Section 3 provides related work. The Boolean algebra concepts and its usage in the sequence matching method are provided in Section 4. Section 5 discusses the concepts of fuzzy logic and its usage in the second method being compared. Section 6 details the classical Needleman-Wunsch algorithm. The algorithms of the both the methods are described in Section 7. Section 8 analyses the time and space efficiency of both the approaches. Experimental results and their discussions are presented in Section 9 and finally Section 10 concludes the paper.

2. SEQUENCE ALIGNMENT

Any biological sequence is a sequence of characters drawn from an alphabet. For DNA sequence, character set is {A, C, G, T}, for RNA sequence, the set is {A, C, G, U}, and for protein sequence, character set is {A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V}. A sequence alignment is the process of identifying one-to-one correspondence among sub-units of sequences in order to measure the similarities among them. The similar regions of the aligned sequences provide functional, structural, and evolutionary information about the sequences under study. Generally, aligned sequences are represented as rows within a matrix. In order to align the identical or similar characters in successive columns, gaps ('-')

are inserted between the characters, Gaps are called indels, as they represent insertion of a character in or a deletion of a character from a biological sequence. Pair-wise sequence alignment is the alignment of two biological sequences. Multiple sequence alignment is the alignment of more than two biological sequences. [12]

Two approaches to sequence alignment: global alignment and local alignment. Local alignments (Fig. 1) identify regions of similarity within long sequences that are often widely divergent overall. Global alignment (Fig. 2) "forces" the alignment to span the entire length of all query sequences.

```
TTCTGTGGCTTACGCGAATC
     |||||
-----TGGCT----------
```

Fig. 1: Local alignment of two biological sequences

```
TTCT GAAGCTT -ACGGGATTC
 ||  ||| ||| ||  | ||
GTC -GAA- CTTGAC TGAAT --
```

Fig. 2: Global alignment of two biological sequences

The classical global alignment technique is the Needleman-Wunsch algorithm, which is based on dynamic programming. The Smith-Waterman algorithm is a general local alignment method also based on dynamic programming.

In order to quantify the similarity achieved by an alignment, substitution matrices are used [12]. These matrices contain a value (positive, zero or negative value) for each possible substitution, and the alignment score is the sum of the matrix's entries for each aligned pair. For gaps (indels), a special gap penalty score is used--a very simple one is just to add a constant penalty score for each indel. The optimal alignment is the one which maximizes the alignment score. Commonly used matrices are PAM (Percent Accepted Mutations) matrices, BLOSUM (BLOck SUbstitution Matrix), etc.

3. RELATED WORK

With the exponentially growing biological sequence databases, extensive demands have been put on the implementation of new fast and efficient sequence alignment algorithms. Most of the research work has been intended on primarily providing new algorithms with the main requisite of the meeting the demands of efficient sequence alignment. Researchers have used all the latest techniques with the aim of providing fast and efficient alignment algorithms.

Needleman and Wunsch proposed a algorithm based on dynamic programming for global alignment of two sequences [1]. Smith and Waterman proposed a dynamic programming algorithm to find a pair of segments one from each of two long sequences such that there is no other pair of segments with greater similarity (homology) [2]. In this local alignment algorithm, similarity measure allowed arbitrary length deletions and insertions. Das and Dey proposed a new algorithm for local alignment of DNA sequences [4]. Direct comparison methods were proposed to obtain global and local alignment between the two sequences by Bandyopadhyay et al. [5]. They also proposed an alternate scoring scheme based on fuzzy concept. An iterative progressive alignment method for multiple sequence alignment was designed using new techniques for both generating guide trees for randomly selected sequences as well as for rearranging the sequences in the guide trees by Naznin, Sarker and Essam [10]. Cai, Juedes, and Liakhovitch proposed to combine existing efficient algorithms for near optimal global and local multiple sequence alignment with evolutionary computation techniques to search for better near optimal sequence alignments [3]. Y. Chen et.al introduced a partitioning approach, based on ant-colony optimization algorithm that significantly improved the solution time and quality by utilizing the locality structure of the problem [7]. A hybrid approach of dynamic programming and fuzzy logic was proposed to align multiple sequences progressively by Nasser et al. [8]. They computed optimal alignment of subsequences based on several factors such as quality of bases, length of overlap, gap penalty. An algorithm for global alignment between two DNA sequences using Boolean algebra was suggested by Anitha and Poorna [11]. They compared the performance of the algorithm with Needleman-Wunsch algorithm. Yue and Tang applied the divide-and-conquer strategy to align three sequences so as to reduce the memory usage from O (n3) to O(n2). They used dynamic programming so as to guarantee optimal alignment [9]. Chang et al. established fuzzy PAM matrix using fuzzy logic and then estimated score for fitness function of genetic algorithm using fuzzy arithmetic [6]. Their experimental results evidenced fuzzy logic useful in dealing with the uncertainties problem, and applied to protein sequence alignment successfully.

Of all the algorithms that had been proposed, the main objective of the researchers had been to apply different techniques in order to provide efficient alignment algorithms in terms of time and memory requirements.

4. APPLYING BOOLEAN LOGIC

Boolean algebra (or Boolean logic) is a logical calculus of truth values (true or false), developed by George Boole in the 1840s. In contrast to elementary algebra, which is based on numeric operations multiplication xy, addition x + y, and negation −x, Boolean algebra is customarily based on logical counterparts to those operations, namely conjunction $x\Lambda y$ (AND), disjunction xVy (OR), and complement or negation $\neg x$ (NOT) [15].

The first method, based on Boolean logic, converts the given biological sequences into binary form, so that Boolean logic can be applied to them. The four nucleotides A, C, G and T are represented by 000, 001, 010, and 011 respectively, and the gaps as 100. Exclusive NOR (XNOR) function (see Table I) is a Boolean operator that produces true if both the inputs are same, otherwise false [16]. XNOR function is applied on two sequences encoded as binary strings. In the resultant string, replace the three consecutive ones by 1, otherwise replace by 0. Thus, in the final resultant string, 1 will correspond to a match and 0 to a mismatch.

Table 1: XNOR gate

A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

5. APPLYING FUZZY LOGIC

Fuzzy logic is a form of multi-valued logic that deals with reasoning that is approximate rather than fixed and exact. In contrast with "crisp logic" i.e. Boolean logic, where binary sets have two-valued logic: true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1 [18]. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. It is based on the fuzzy-set theory proposed by L.A. Zadeh in 1965.

In a fuzzy system, the values of a fuzzified input execute all the rules in the knowledge repository that have the fuzzified input as part of their premise. This process generates a new fuzzy set representing each output or solution variable. Defuzzification creates a value for the output variable from that new fuzzy set [13]. So, in order to apply fuzzy logic to an application, first the inputs must be fuzzified so that their value is in the range 0 to 1, then the rules defined by the application are applied, and after this, the results derived from various rules are combined using an aggregation function. Finally, the aggregated results are defuzzified by using an inference function. The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations. The operations on fuzzy sets are different than the operations on non-fuzzy sets [14]. The operations for OR and AND operators are max and min, respectively. For complement (NOT) operation, NOT(A) is evaluated as (1-A).

The second matching technique being discussed uses three input variables – match-count (#match), mismatch-count (#mismatch), and calculated-score (#score – calculated using substitution matrix). These inputs are then fuzzified using following membership functions:

$$\mu(\text{match}) = \left\{ \begin{array}{l} 0, \text{ if } \#\text{match}=0 \\ 1, \text{ if } \#\text{match}=\text{lenSeq} \\ [0,1]\ (1 - \#\text{match} / \text{lenSeq}) \end{array} \right\} \qquad - (1)$$

$$\mu(\text{mismatch}) = \left\{ \begin{array}{l} 0, \text{ if } \#\text{mismatch}=0 \\ 1, \text{ if } \#\text{mismatch}=\text{lenSeq} \\ [0,1]\ (1 - \#\text{mismatch} / \text{lenSeq}) \end{array} \right\} \qquad - (2)$$

$$\mu(\text{score}) = \left\{ \begin{array}{l} 0, \text{ if } \#\text{score} <=0 \\ 1, \text{ if } \#\text{score} = \text{perfectScore} \\ [0,1]\ \#\text{score} / \text{perfectscore} \end{array} \right\} \qquad - (3)$$

In these equations, lenSeq is the length of the shorter sequence of the two sequences being matched, and perfectScore is the score of matching the two candidate sequences, if there are no indels or replacements.

6. THE NEEDLEMAN-WUNSCH ALGORITHM

The Needleman-Wunsch algorithm is a classical dynamic programming- based algorithm for global alignment of two biological sequences. This algorithm first calculates a scoring matrix for the two given sequences A and B, by placing one sequence along row side and another column side. The size of the matrix is (M+1)*(N+1) (M and N are the lengths of the two sequences). The optimal score at each matrix (i, j) position is calculated by adding the current match score to previously scored positions and subtracting gap penalties, which may evaluate to either a positive, negative or 0 value.

A matrix F(i, j) indexed by residues of each sequence is built recursively, such that

$$F(i, 0) = F(0, j) = 0$$

$$F(i, j) = \max \left\{ \begin{array}{l} F(i-1, j-1) + S(x_i, y_j), \\ F(i-1, j) + G, \\ F(i, j-1) + G \end{array} \right\}$$

subject to boundary conditions; here, S(i, j) is the substitution score for residues i and j, and G is the gap penalty [17].

The methods discussed in the paper use the Needleman-Wunsch algorithm for aligning two biological sequences. The substitution matrix S, given in Figure 3, is used to calculate the scores.

	A	C	G	T
A	2	-1	1	-1
C	-1	2	-1	1
G	1	-1	2	-1
T	-1	1	-1	2

Fig. 3: Substitution Matrix used in the algorithm

An alignment is computed using the F-matrix (calculated above): start from the bottom right cell, and compare the cell value with the three possible sources ((i-1, j-1) i.e. a Match, (i, j-1) i.e. an Insert, and (i-1, j) i.e. a Delete) to see which it came from. If it is same as Match, then Ai and Bj are aligned, if same as Delete, then Ai is aligned with a gap, and if same as Insert, then Bj is aligned with a gap.

7. ALGORITHM

In this paper, we compare two methods of biological sequence matching. Both these methods align multiple sequences (DNA) progressively using the algorithm ALIGN_SEQ. Both the methods differ in the way they calculate the similarity of the two sequences. The first method employs Boolean algebra to calculate the percentage of similarity of the two sequences. The algorithm used in this method is BA_MATCH_SEQ. The second method uses fuzzy logic to determine the similarity value of the two sequences. The algorithm FL_MATCH_SEQ is used by the second method. The three algorithms are discussed below:

Algorithm BA_MATCH_SEQ (A, B)

This algorithm finds the percentage of similarity (Match-Score) of the given two sequences A and B. The flowchart for the algorithm is given in Fig. 4.

Algorithm FL_MATCH_SEQ (A, B)

This matching algorithm finds the fuzzy score of similarity, based on fuzzy parameters (as given in section 5), for the given two biological sequences A and B. The corresponding flowchart is given in Fig. 5. The weights W1, W2, and W3 are the fixed weights assigned to three fuzzy measures respectively.

Algorithm ALIGN_SEQ (SeqDB, N)

This algorithm aligns the given set of N sequences SEQDB, progressively, and stores the aligned sequences in AlignedSeqDB. The algorithm follows progressive approach to multiple sequence alignment, by first aligning the two most similar sequences, using Needleman-Wunsch algorithm. Then picking the sequences, one by one, from the rest, and aligning it to the aligned set of sequences. This algorithm uses the algorithm MATCHER, which is algorithm BA_MATCH_SEQ in case of first method, and FL_MATCH_SEQ in case of second method. The flowchart is given in Fig. 6

8. COMPARATIVE ANALYSIS OF APPROACHES

Both the methods being compared follow the same progressive approach to align multiple biological sequences, but they differ in the way they calculate the similarity score.

The Boolean algebra method requires more memory than the fuzzy logic method. The former method requires auxiliary memory to store the binary forms of the candidate sequences, whereas no such memory requirement is there in the later method. The amount of auxiliary memory depends on the language being used for implementation. Like in MATLAB, a character takes 2 bytes whereas a logical field requires 1 byte. So, in this case, the extra memory required for binary biological sequences is (3*M/2), where M is the memory required by character biological sequences.

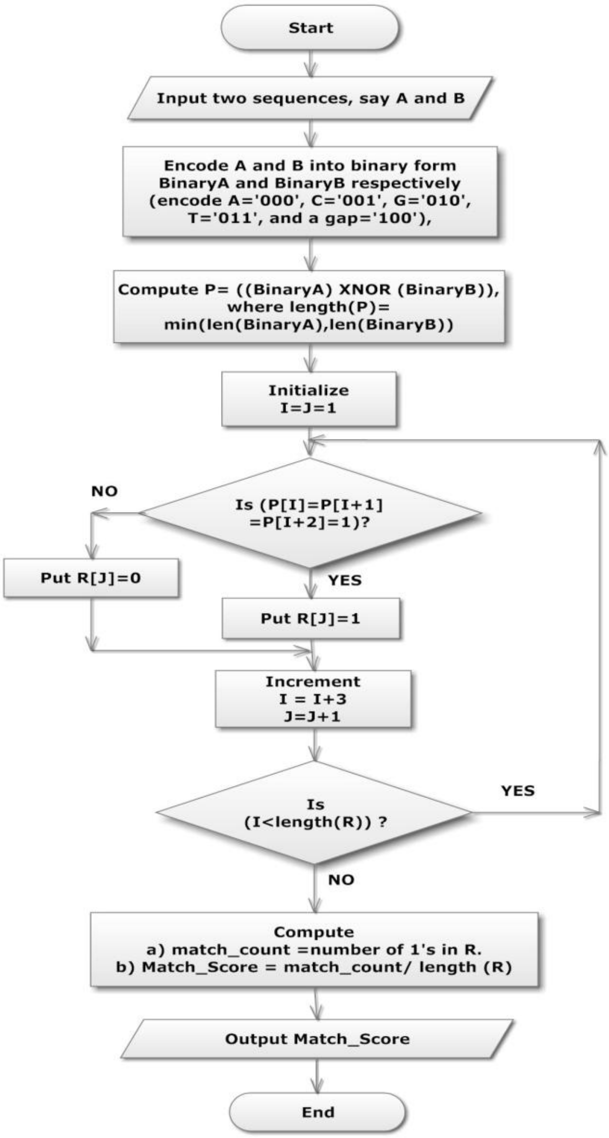
Start
Input two sequences, say A and B
Encode A and B into binary form BinaryA and BinaryB respectively (encode A='000', C='001', G='010', T='011', and a gap='100'),
Compute P= ((BinaryA) XNOR (BinaryB)), where length(P)= min(len(BinaryA),len(BinaryB))
Initialize I=J=1
Is (P[I]=P[I+1] =P[I+2]=1)?
NO
Put R[J]=0
YES
Put R[J]=1
Increment I = I+3 J=J+1
Is (I<length(R)) ?
YES
NO
Compute a) match_count =number of 1's in R. b) Match_Score = match_count/ length (R)
Output Match_Score
End

Fig. 4: Flowchart for sequence matching algorithm (BA_MATCH_SEQ) using Boolean algebra approach.

The Boolean algebra method also requires more computation time than the fuzzy logic method. The extra time is required for converting the given character sequences to binary sequences in case of former method. No such conversion is there, in case of later method. If L is an average length of input sequences, then

Boolean algebra method: Time required to
Convert two sequences into binary form = 2*L
Perform XNOR operation=3*c1*L, where c1 is time to perform XNOR operation on one bit
Compute resultant string=c2*L, where c2 denotes time to compare the bits
Compute match value=c3*L, where c3 is time to count 1s in resultant string
Total Time= (L+(3*c1+c2+c3)*L) units

Fuzzy Logic method: Time required to
Compute #match, #mismatch, and #score =L
Fuzzify inputs= 3* t1, t is a constant to calculate one equation
Compute aggregate fuzzy value=t2, a constant time
Compute defuzzified value=t3, a constant time
Total Time= (L+t1+t2+t3) units

Ignoring the constants, we see that the time required by Boolean method is more than the time required by fuzzy method; the constants in Boolean method time depend on the length of the sequences whereas this is not the case in Fuzzy method. From this, we deduce that fuzzy method is efficient in terms of time and memory requirements.

9. EXPERIMENTAL RESULTS AND DISCUSSIONS

The algorithms for both the methods were implemented using MATLAB^{TM}. The fuzzy logic GUI tool provides a convenient way to build and implement fuzzy inference systems [19]. We used this tool to design a fuzzy sequence matcher system for implementing the fuzzy method of sequence matching (Fig. 7).

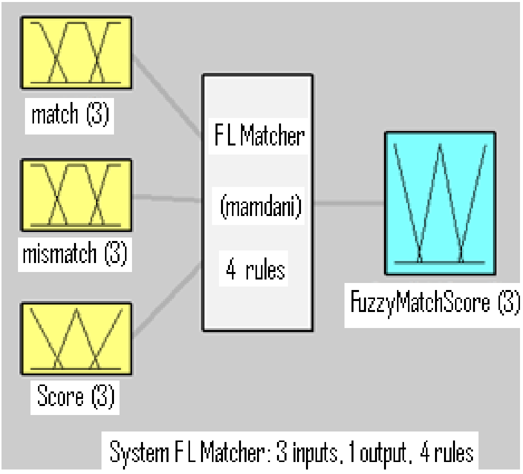

Fig. 7: Fuzzy Inference System FL_Matcher with 3 inputs, 1 output, and 4 rules, designed using MATLAB Fuzzy Logic toolbox

Seven different sets of influenza virus genome sequences: AH1N1, AH1N2, AH1N3, AH2N1, AH2N2, AH3N1 and AH3N2, for different countries were collected (randomly) from NCBI's Influenza virus resource site [20]. Details of the tested sequence sets are listed in Table 2.

Table 2: Tested sequence sets

Influenza Virus type (DNA))	No. of sequences	Length Range (bp)	Origin (Country/ Continent)	Collection Year
AH1N1	38	1070 to 1701	Japan	2002 -2010
AH1N2	13	1003 to 1744	Japan	1980 -2010
AH1N3	12	1651 to 1752	Mixed	2002 - 2009
AH2N1	10	1176 to 1747	Mixed	1986 - 2003
AH2N2	40	1017 to 1741	North America	1957 - 2005
AH3N1	23	707 to 1746	Mixed	2000- 2010
AH3N2	55	982 to 1021	Japan	2002 - 2010

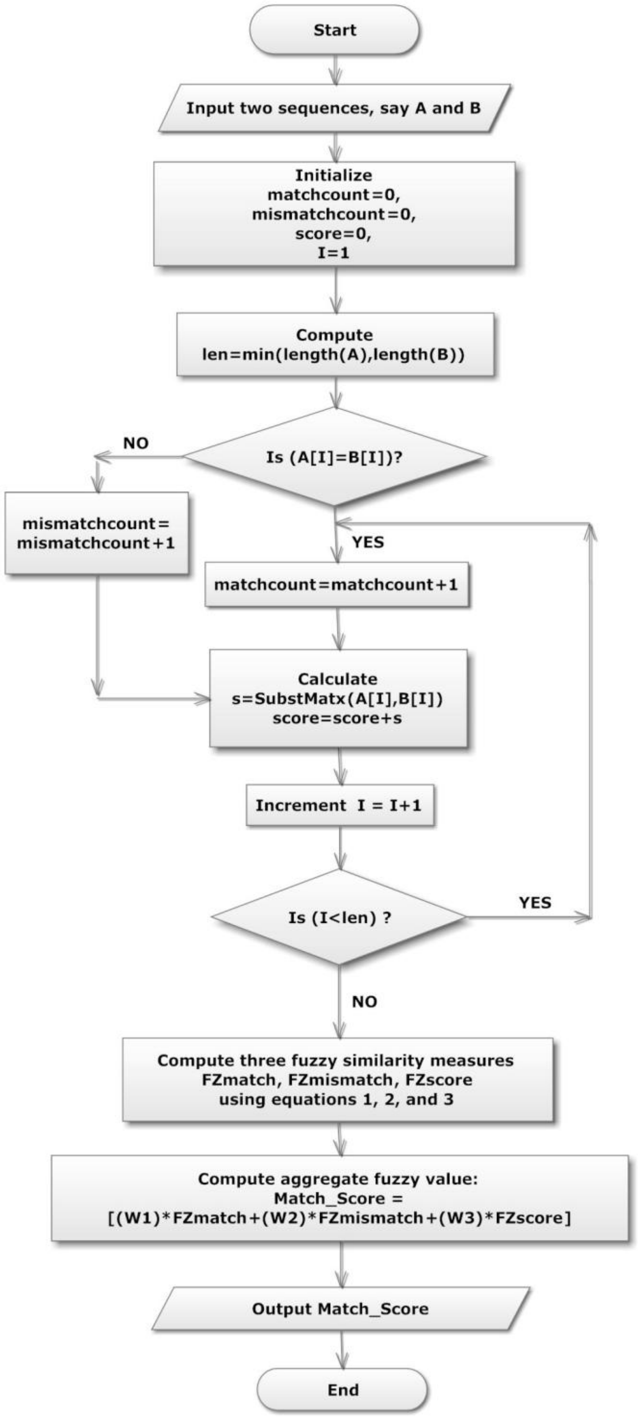

Fig. 5: Flow-chart for sequence matching algorithm (FL_MATCH_SEQ) using fuzzy logic approach

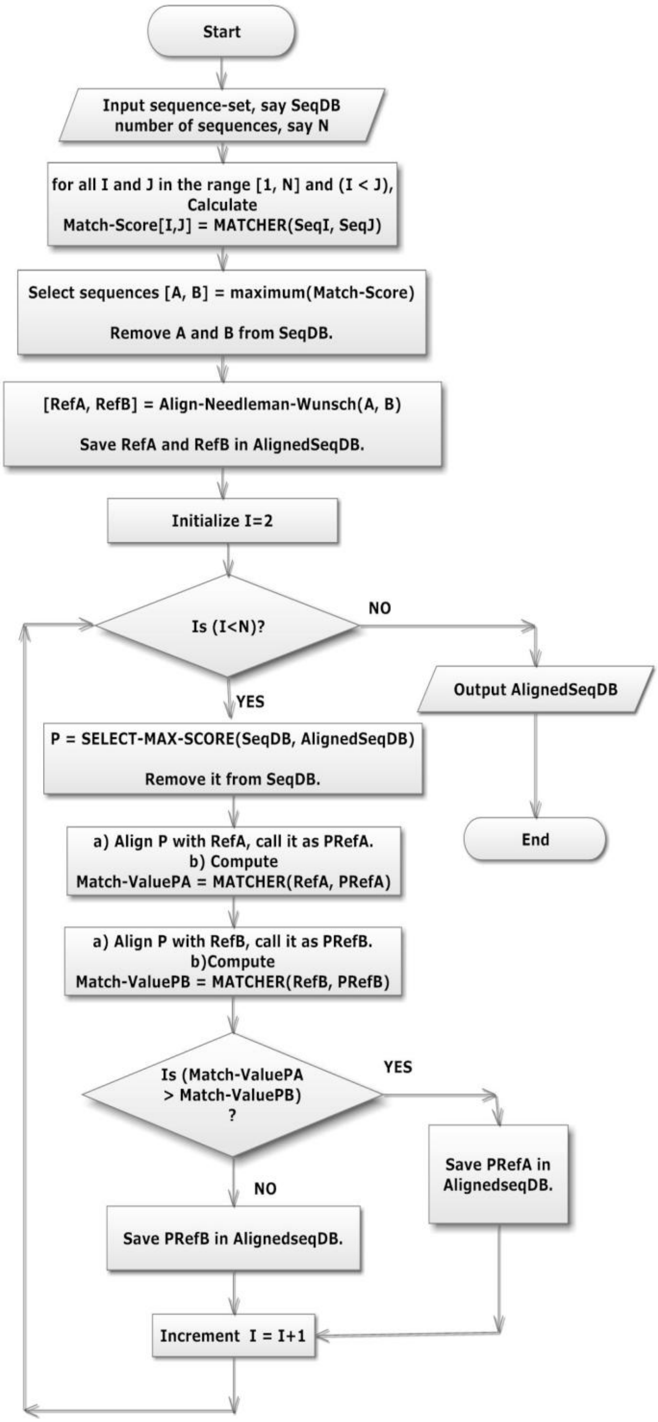
Start
Input sequence-set, say SeqDB
number of sequences, say N
for all I and J in the range [1, N] and (I < J),
Calculate
Match-Score[I,J] = MATCHER(SeqI, SeqJ)
Select sequences [A, B] = maximum(Match-Score)
Remove A and B from SeqDB.
[RefA, RefB] = Align-Needleman-Wunsch(A, B)
Save RefA and RefB in AlignedSeqDB.
Initialize I=2
Is (I<N)?
NO
YES
Output AlignedSeqDB
End
P = SELECT-MAX-SCORE(SeqDB, AlignedSeqDB)
Remove it from SeqDB.
a) Align P with RefA, call it as PRefA.
b) Compute
Match-ValuePA = MATCHER(RefA, PRefA)
a) Align P with RefB, call it as PRefB.
b)Compute
Match-ValuePB = MATCHER(RefB, PRefB)
Is (Match-ValuePA
> Match-ValuePB)
?
YES
NO
Save PRefA in
AlignedseqDB.
Save PRefB in AlignedseqDB.
Increment I = I+1

Fig. 6: Flow-chart for progressive alignment of sequences (ALIGN_SEQ)

These seven sets of sequences were aligned using both the techniques, in discussion, and also with the CLUSTALW algorithm. CLUSTALW is a general purpose global multiple sequence alignment tool for proteins and DNA; it employs progressive alignment based on Needleman-Wunsch algorithm. To align the test-sequences with CLUSTALW, we have used web-based version of CLUSTALW, available at GenomeNet [21]. We used the SP (Sum-of-Pairs) score and CS (Column Score) as measures to compare the performances of the three methods. The SP score is calculated using the substitution matrix given in Fig. 3 and with gap penalty as 2. The CS is defined as the number of correctly aligned columns in the alignment divided by the length of the reference alignment. The closer to 1.0 these scores are, the better the alignment is. The SP score and column scores of the alignment results produced by three methods for seven test- sequence sets are tabulated in Table 3. The results are plotted as bar charts in Fig. 8a and 8b.

The alignment results for the proposed methods – Boolean algebra and fuzzy logic, indicate that their performance is comparable with the popular CLUSTALW algorithm. The results also reveal that the two methods being compared do equally well as the scores for both the methods are almost the same, except for one data set – AH2N1 virus DNA sequences. This difference can be accounted for high dissimilarity due to different origin of the sequences and different collection years. The two data sets – AH1N3 and AH3N1, have mixed origin like AH2N1, but their collection time is 2000 onwards, whereas AH2N1 data set is 1986-2003. This indicates that Boolean method is not as good as the fuzzy method when the involved sequences' dissimilarity is high.

From the above discussion, we can say that the alignment results produced by the two methods are comparable to the well-known CLUSTALW algorithm, but they differ in time and space requirements. The Boolean method is very inefficient if the number and length of involved sequences is large, and even if the involved sequences are highly dissimilar. The fuzzy method has no such restrictions.

Influenza Virus Type (DNA sequences)	Boolean Algebra Method		Fuzzy Logic Method		CLUSTALW	
	SP Score	CS	SP Score	CS	SP Score	CS
AH1N1	0.4089	0.7037	0.4089	0.7037	0.4089	0.7037
AH1N2	0.9278	0.9002	0.9304	0.9019	0.9333	0.903
AH1N3	0.8136	0.8213	0.8136	0.8239	0.8128	0.8196
AH2N1	0.235	0.5516	0.2408	0.5567	0.2371	0.5496
AH2N2	0.9601	0.9523	0.9601	0.9523	0.9601	0.9523
AH3N1	0.859	0.8441	0.859	0.8441	0.8584	0.8429
AH3N2	0.9143	0.951	0.9143	0.951	0.9143	0.951

Table 3: SP scores and Column scores of alignments of seven influenza virus type DNA sequence sets by three different methods

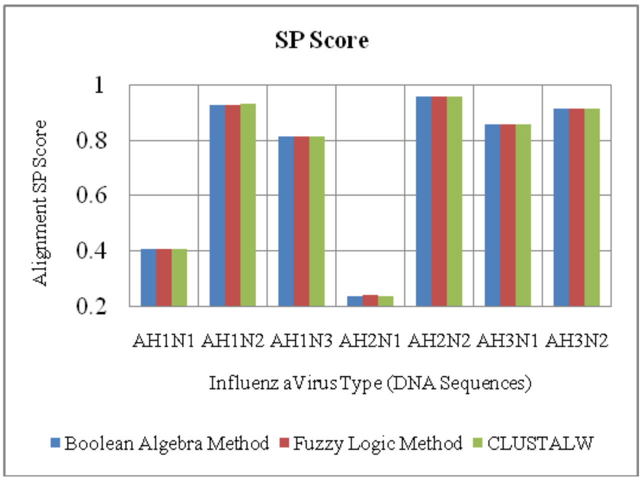

Fig. 8a: Graph for SP score of alignments of various DNA sequence sets with three methods: Boolean algebra, Fuzzy logic, and CLUSTALW.

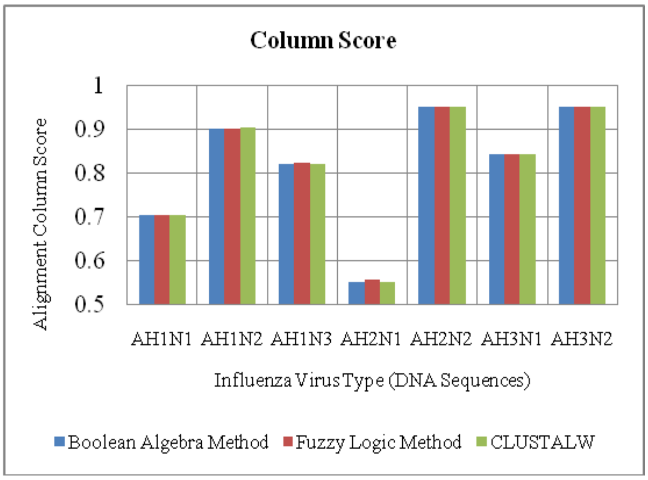

Fig. 8b: Graph for Column score of alignments of various DNA sequence sets with three methods: Boolean algebra, Fuzzy logic, and CLUSTALW.

10. CONCLUSION

Two methods for sequence matching: Boolean algebra and fuzzy logic have been discussed, implemented and tested on real data sets. In the first method, the sequences were encoded into binary form and then using logical operators, the match value was determined. In the second method, number of matches, mismatches and the score for two sequences were determined, and then fuzzified so as to apply fuzzy logic to calculate the fuzzy similarity value. In the both methods, the match/ similarity value guides in the progressive alignment of multiple sequences, which was done using dynamic programming. The experimental results show that both the algorithms perform the alignment of sequences well, and their performances are comparable with the popular CLUSTALW algorithm. The small differences in their performances can be reasoned for differences in origin and collection time of the test-sequences. The Boolean method is not as accurate as the fuzzy method if the dissimilarity among the sequences to be aligned is large. The analysis of both the methods shows that Boolean method suffers from time and space inefficiency, as compared to the fuzzy logic method. This means that Boolean approach is not viable if the number and the length of the sequences being aligned is large, and its performance deteriorates if the involved sequences have high degree of dissimilarity, but the fuzzy approach is not bound to any such constraints.

11. REFERENCES

[1] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Molecular Biology*, vol. 48, pp. 443-453, 1970.

[2] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequence," *J. Molecular Biology*, vol. 147, pp. 195-197, 1981.

[3] L. Cai, D. Juedes, E. Liakhovitch, "Evolutionary computation techniques for multiple sequence alignment", *Proceedings of the 2000 Congress on Evolutionary Computation*, 2000, pp. 829-835

[4] Swagatam Das & Debangshu Dey, "A new algorithm for local alignment in DNA sequencing", *Proc. of IEEE Conference, INDICON 2004*, pp. 410-413.

[5] Bandyopadhyay, S.S.; Paul, S.; Konar, A., “Improved Algorithms for DNA Sequence Alignment and Revision of Scoring Matrix”, Proceedings of International Conference on Intelligent Sensing and Information Processing, 2005, pp. 485-490

[6] Pin-Teng Chang, Lung-Ting Hung, Kuo-Ping Lin, Chih-sheng Lin, Kuo-Chen Hung, “Protein Sequence Alignment Based on Fuzzy Arithmetic and Genetic Algorithm”, 2006 IEEE International Conference on Fuzzy Systems, pp. 1362-1367

[7] Y. Pan, Y. Chen, Juan Chen, Wei Liu, Ling Chen “Partitioned optimization algorithms for multiple sequence alignment”, Proceedings of the 20th International Conference on Advanced Information Networking and Applications, 2006, pp. 5

[8] Sara Nasser, Gregory L. Vert, Monica Nicolescu1 and Alison Murray, “Multiple Sequence Alignment using Fuzzy Logic”, Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 304-311

[9] Feng Yue and Jijun Tang, “A Divide-and-Conquer Implementation of Three Sequence Alignment and Ancestor Inference”, 2007 IEEE International Conference on Bioinformatics and Biomedicine, pp. 143-150

[10] Farhana Naznin, Ruhul Sarker, and Daryl Essam , “Iterative Progressive Alignment Method (IPAM) for Multiple Sequence Alignment”, Computers & Industrial Engineering, 2009. pp. 536-541

[11] V. Anitha and B. Poorna, “DNA Sequence Matching using Boolean Algebra”, 2010 International Conference on Advances in Computer Engineering, pp. 212-216.

[12] David W. Mount, “Bioinformatics: Sequence and Genome Analysis”, Cold Spring Harbor Laboratory Press

[13] E. Cox, “Fuzzy Fundamentals”, IEEE Spectrum October 1992, Volume 29, Issue 10, pp 58-61.

[14] “A Short tutorial on Fuzzy Logic”, http://www.cs.bilkent.edu.tr/~bulbul/depth/fuzzy.pdf

[15] Wikipedia Website, “Boolean Algebra”, http://en.wikipedia.org/wiki/Boolean_algebra_(logic).

[16] Wikipedia Website, “XNOR gate”, http://en.wikipedia.org/wiki/XNOR_gate.

[17] Wikipedia Website, “Needleman-Wunsch Algorithm”, http://en.wikipedia.org/wiki/Needleman-Wunsch_ algorithm.

[18] Wikipedia Website, “Fuzzy Logic”, http://en.wikipedia.org/wiki/Fuzzy_logic.

[19] Mathworks Website, “Fuzzy Logic Toolbox”, http://www.mathworks.com/help/toolbox/fuzzy.

[20] NCBI Website, “Influenza Virus Data base”, http://www.ncbi.nlm.nih.gov/genomes/FLU/Database/nph-select.cgi.

[21] GenomeNet Tools Website, “CLUSTALW TOOL”, http://www.genome.jp/tools/clustalw.