ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

HMM_MLCS: Hidden Markov Model (HMM) based algorithm to identify Multiple Longest Common Subsequence (MLCS) in DNA Sequences

[1]B. Devika Rubi*, [2]Dr. L. Arockiam
[1]Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore
[2]Associate Professor, Department of Computer Science, St Joseph's College, Tiruchirappalli
***Corresponding author: E-Mail: deviraja@gmail.com**

ABSTRACT

Multiple Longest Common Subsequence (MLCS) refers to find the Longest Common Subsequence between two or more sequences. Identifying MLCS in DNA sequences is helpful to generate Phylogenetic tree, Motif identification and DNA sequence alignment. The existing Dynamic Programming based MLCS algorithms require exponential time and space complexity. The statistical method Hidden Markov Model (HMM) helps to identify highly aligned sequences. MLCS identification is nothing but identifying the longest aligned subsequence among the DNA sequences. This paper proposes a HMM based MLCS algorithm for DNA sequences. The proposed HMM_MLCS identifies MLCS with linear time and space complexity.

KEY WORDS: Longest Common Subsequence, Hidden Markov Model (HMM), Dynamic Programming, Aligned Sequences.

1. INTRODUCTION

DNA sequences are the linear arrangements of the four chemicals namely Adenine (A), Thymine (T), Cytosine (C) and Guanine (G) in any order. New DNA sequences are found by the existing ones. The existing sequences can transfer information about structure/functionalities into the new sequences. If two sequences are related, then they are called as homologous/alignment. Multiple Longest Common Subsequence (MLCS) (Hirschberg, 1975; 1977; Rick, 1994; Kumar and Rangan, 1987) refers to find the Longest Common Subsequence between two or more sequences. Identifying MLCS is useful to find homologous between DNA sequences. Homologous sequences are helpful to generate Phylogenetic tree, Motif identification and DNA drug design (Trifonov and Berezovsky, 2003; Sankoff, 1972; Dayhoff, 1969).

The statistical method, HMM (Fujiwara, 1994; Rabiner and Juang, 1986; Stolcke and Omohundro, 1993) is used to model a sequence or a family of sequences. HMM is useful for sequence alignment (Smith and Waterman, 1981; Vingron, 1996). HMM model generate current character of the sequence with respect to the probability of the previous character of the sequence. This paper discusses about a HMM based MLCS algorithm for DNA sequences. The existing Dynamic Programming based MLCS require exponential time and space complexity. The proposed HMM_MLCS identifies MLCS with linear time and space complexity.

This paper is organized as given below. Section 2 defines MLCS materials and discusses about various existing methods with their time and space complexities. Section 3 proposes a new algorithm called HMM_MLCS() to identify MLCS with an illustration. Section 4 discusses about the implementation and analyses the proposed algorithm HMM_MLCS(). Section 5 provides the conclusion and future research direction.

2. MATERIALS AND METHODS

2.1. MLCS Problem Definition: A sequence $Z = < z_1, z_2 \ldots z_n >$ is called Multiple Longest Common Subsequence (MLCS) of other sequences $A = < a_1, a_2 \ldots a_n >$, $B = < b_1, b_2 \ldots b_n > \ldots\ldots$ $K = < k_1, k_2 \ldots k_n >$ and A, B….K are the super sequences of Z denoted as $Z \subseteq \{A, B, \ldots. K\}$, if there exists integers $1 \leq j_1 \leq j_2 \ldots \leq j_n \leq m$ such that $Z_1 \subseteq \{ a_{j1}, b_{j1}, c_{j1} \ldots k_{j1}\}$, $Z_2 \subseteq \{ a_{j2}, b_{j2}, c_{j2} \ldots k_{j2}\}$ $\ldots.Z_n \subseteq \{ a_{jn}, b_{jn}, c_{jn} \ldots k_{jn}\} \leq m$

Thus MLCS is a longest common subsequence of more than two sequences where event e_1 occurs before e_2, e_2 occurs before e_3, etc. Let $A = a_1 a_2 a_3 \ldots a_m$ and $B = b_1 b_2 \ldots\ldots b_n$ are the two sequences. And 'Z' is the Longest Common Subsequence (LCS) between A and B, which is defined as $z_1 z_2 \ldots z_k$.

Table.1.Sample DNA sequences

Sequence / Position#	1	2	3	4	5	6	7	8	9	10
A	C	T	G	C	T	C	A	C	G	C
B	C	A	A	C	T	C	T	C	A	C

The LCS of sample DNA sequences in Table 1 is "C T C A C".

2.2. Existing LCS Methods: The major three existing methods to identify MLCS are a) Dynamic Programming Method (DP), b) Dominant Point Method, c) Cache-Oblivious Method.

2.2.1. Dynamic Programming Method: Dynamic programming (DP) method ((Akutusu, 2000), (Apostolico, et al., 1992), (Masek & Paterson, 1980) and (Bentley, 1980)) defines the current stage from the previous stage. The score matrix for DP method is defined in equation.

$$L[i,j] = \begin{cases} 0 & if \quad i \; or \; j = 0, \\ L[i-1, j-1] + 1, & if \;\; s_1[i] = s_2[j] \\ max(L[i, j-1], L\,[i-1, j]), & if \;\; s_1[i] \neq s_2[j] \end{cases}$$

The score matrix for the sample DNA sequences in Table 1 is shown in Table 2.

Table.2.Score Matrix L[i,j] for sample Data

		j ↓ 0	j ↓ 1	j ↓ 2	j ↓ 3	J ↓ 4	j ↓ 5	j ↓ 6	j ↓ 7	j ↓ 8	J ↓ 9	j ↓ 10
			C	A	A	C	T	C	T	C	A	C
i→0		0	0	0	0	0	0	0	0	0	0	0
i→1	C	0	**1**	1	1	1	1	1	1	1	1	1
i→2	T	0	1	1	1	1	**2**	2	2	2	2	2
i→3	G	0	1	1	1	1	2	2	2	2	2	2
i→4	C	0	1	1	1	2	2	3	3	3	3	3
i→5	T	0	1	1	1	1	3	3	4	4	4	4
i→6	C	0	1	1	1	2	2	4	4	5	5	5
i→7	A	0	1	2	2	2	2	4	4	5	6	6
i→8	C	0	1	2	2	3	3	3	4	5	6	7
i→9	G	0	1	2	2	3	3	3	4	5	6	7
i→10	C	0	1	2	2	3	3	4	4	5	6	7

The identification of MLCS by using DP method contains two steps. In the first step score matrix is formed. Subsequently, in the second step score matrix is traced to identify the required MLCS. Thus, if "n" is the length of the sequences and "d" is the number of sequences then time and space complexity to identify MLCS is $O(n^d)$.

2.2.2. Dominant Point Method: The score matrix defined in Eq. 1 shows that MLCS occurs at the first occurrence of matching character's position in the score matrix. These positions are called as dominant points. i.e. MLCS is the subset of the dominant points. Instead of tracing the entire score matrix to find MLCS it is enough to find the subset of dominant point set ((Wang, et al., 2011), (Hakata & Imai, 1998) and (Kung, et al., 1975)). This reduces the time and space complexity to identify MLCS.

Dominant points set D, for the sample DNA sequences in Table 1 is { (1,1) (2,5), (4,4), (4,6), (5,7), (6,6), (6,8), (7,9), (8,10)}. The subset of "D" is {(4,6), (5,7), (6,8), (7,9), (8,10)}is the required MLCS, by eliminating event which are not following the order of $e_1 < e_2 \ldots < e_k$. i.e. for instance (2,5) is not less than (4,4).

If "d" is the number of sequences, "n" is the length of sequences, "D" is the size of dominant point set, and "N" is the number of levels, then the Time complexity is O (d N $\log^{d-2}$ n). And the space complexity of this algorithm is O (| D | d + n | $\sum$ | d).

2.2.3. Cache Oblivious methodology: The main difficulty in MLCS identification is transfer of large number of sequence data between main and cache memory. This delays the execution time. This method recursively apply divide and conquer method on score matrix by keeping track of the boundary positions of the sub-matrices. This reduces the transfer rate of data between main and cache memory (Chowdhury, 2007)).

If 'n' is the length of sequence and 'd' is the number of sequences, then the time complexity of the Cache Oblivious DP algorithm is O(n^d) and the space complexity is O (n^{d-1}). The time and space complexity of this method is exponential.

2.3. HMM model and the proposed HMM_MLCS Algorithm

2.3.1. Hidden Markov Model (HMM): The statistical method, HMM is defined by

a) A set of states Q

b) A set of transitions, where transition probability

$a_{kl} = P(\pi_i = l \,/\, \pi_{i-1} = k)$, is the probability of transitioning from state k to state l for k_i l ε Q

c) An emission probability,

$e_k(b) = P(x_i = b \,/\, \pi_1 = k)$, for each state k and each symbol b where $e_k(b)$ is the probability of seeing b in state k.

The sum of all emission probabilities at a given state must equal to 1, ie. $\sum_b e_k = 1$for each state k. The HMM model helps to identify highly aligned sequences where the log_odd_ratio of highly aligned sequences should be closer to 0. As MLCS identification is nothing but identifying the longest aligned subsequence among the DNA sequences. Thus HMM model helps to identify MLCS in a linear time complexity.

Two sample DNA sequences considered for our illustration are listed out in Table 3.

Table.3.Sample DNA sequences

Positions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Seq1	T	A	A	T	C	G	A	A	C	T	A	C	A	G	G	A
Seq2	A	T	C	G	G	A	T	C	A	T	A	T	C	G	C	C

Positions	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Seq1	T	A	G	A	T	C	G	A	A	T	G	G	T	G	G
Seq2	G	A	A	C	T	A	C	A	G	G	T	T	A	A	C

The HMM model for the Seq_1 in Table 3 is as shown in Figure 1. Transition probabilities of possible sixteen len_2 sub_patterns for Seq_1 in Table 3 are listed out in Table 4. The HMM model for the Seq_2 in Table 3 is as shown in Figure 2. Transition probabilities of possible sixteen len_2 sub_patterns of Seq_2 in table 3 are listed out in Table 5. The sum of emission probability for the possible 16 states is1.

2.3.2. Proposed HMM_MLCS algorithm: The proposed HMM_MLCS algorithm contains three steps (i) calculate the possible sixteen len_2 probabilities for the DNA sequences (ii) calculate the log_odd_ratio, i.e. the alignment ratio for all sub_patterns of assumed len_10 (iii) lists out the possible MLCS or highly aligned sub_patterns whose log_odd_ratio < 0.5. The pseudo code for proposed HMM_MLCS is as shown in Figure 3.

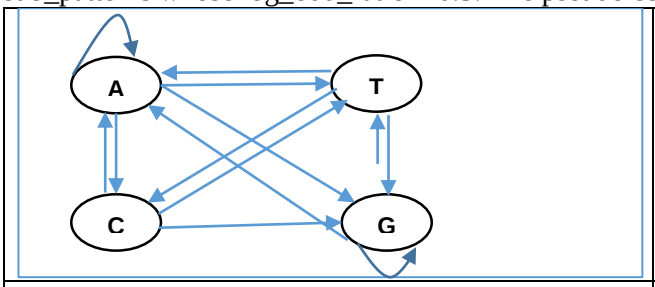

Fig.1.HMM Model for Seq_1

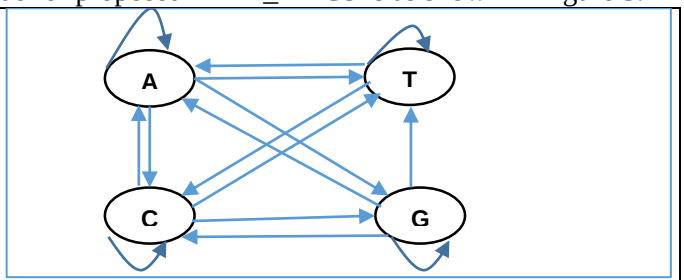

Fig.2.HMM Model for Seq_2

```
//HMM_MLCS algorithm to find LCS
Procedure HMM_MLCS()
{
// s1, s2 are the sequences of length_n
PS1_AA, PS1_AT, PS1_AG, PS1_GG are the 16 possible probabilities for the length_2 pattern in Seq_1
 PS2_AA, PS2_AT, PS2_AG,  PS1_GG are the 16
possible probabilities for the length_2 pattern in Seq_2
Splitting the sequence s1 into consecutive sub_patterns X, of  assumed len_10
For i = 0 to n-1
{
X = s1.substring (i, i+9);
// Calculate log_odd_ratio for X, by splitting X into
// len_2 sub_patterns (pattern_len_2)
For j = 0 to 8
{
pattern_len2 = X.substring(j, j+1);
log_odd_ratio = 0;
switch (pattern_len2)
{
Case “AA” :
{ y = log (PS1_AA/PS2_AA);
log_odd_ratio = log_odd_ratio + y;
break;}
Case “AT” :
{ y = log (PS1_AT/PS2_AT);
log_odd_ratio = log_odd_ratio + y;
break;}
Case “AC” :
{ y = log (PS1_AC/PS2_AC);
log_odd_ratio = log_odd_ratio + y;
break;}
Case “AG” :
{ y = log (PS1_AG/PS2_AG);
log_odd_ratio = log_odd_ratio + y;
break;}
Case “GG” :
{ y = log (PS1_GG/PS2_GG);
log_odd_ratio = log_odd_ratio + y;
break;}
} End Switch case
} End for j
//Identifying MLCS such that log_odd_ratio of MLCS
// closer to 0
If ( log_odd_ratio < 0.5) {
Print (“possible MLCS:” X)
Log_odd_ratio = 0;
} end if
} end for  i
} End HMM_MLCS
```

Fig.3.Pseudo code for proposed HMM_MLCS

Table.4.Transition probabilities of seq_1

	A	T	C	G
A	0.1	0.13	0.06	0.06
T	0.1	0	0.06	0.06
C	0.03	0.03	0	0.06
G	0.13	0.03	0	0.1

Table.5.Transition probabilities of seq_2

	A	T	C	G
A	0.06	0.13	0.1	0.03
T	0.1	0.03	0.1	0
C	0.06	0.03	0.03	0.1
G	0.06	0.03	0.03	0.06

The proposed HMM_MLCS applies the HMM model for the DNA states Q = { A, T, C, G } and the possible sixteen len_2 transition states S = { AA, AT, AC, AG….. GG }. As DNA sequences are the linear arrangement of A, T, C, G in any order, there will be only 16 (=2^4) possible combinations (Permutations with Repetition) of len_2 sub_patterns.

3. ANALYSIS OF HMM_MLCS

3.1. Illustration of HMM_MLCS algorithm: In this section, the proposed HMM_MLCS has been illustrated for the sample MLCS pattern X = { CGAACTACAGG } with the sample DNA sequences in table 3. The consecutive len_2 sub_patterns of X and their log_odd_ratio values are listed in Table 6.

Table 6: Consecutive len_2 patterns of X and log_odd_ratio

len_2 patterns	CG	GA	AA	AC	CT	TA	AC	CA	AG	GG
Log_odd_ratio	-0.2	0.37	0.2	-0.22	-0.22	-0 .3	0.3	0.22	0.37	-0.22

Sum of log_odd_ratio of pattern X is approximately 0.1.

3.2. Time and Space complexity: If "n" is the length of the sequence, then to calculate probability of sixteen sub_patterns of len_2 is (n - 2). If "l" is the assumed length of MLCS, then the number of splitted sub_patterns is (n - l). Each (n – l) patterns require (l – 1) len_2 sub_patterns to calculate log_odd_ratio. Thus, the time complexity is defined as

$$T(n) = (n-2) + ((n-l) * (l-1)), \text{ where } n, l > 0$$

$$= (n-2) + (n\,l - l^2 - l) = O(n),$$ The space complexity of proposed HMM_MLCS is O (n).

3.3. Implementation details and Results: This algorithm has been implemented using Java on a Windows 10 machine with i7 Intel processor 2.33 GHZ, 16 GB RAM. The runtime results of HMM_MLCS algorithm for the given sequences in Table 3 is as shown in Table 7.

Table.7.Runtime results of HMM_MLCS algorithm

PatternID	Position	Pattern	log_odd_ratio
1	1 - 10	TAATCGAACTA	-0.051152522
2	2 - 11	AATCGAACTAC	-0.352182518
3	3 - 12	ATCGAACTACA	-0.227243782
4	4 - 13	TCGAACTACAG	-0.051152522
5	5 - 14	CGAACTACAGG	0.425968732
6	6 - 15	GAACTACAGGA	0.602059991
7	7 - 16	AACTACAGGAT	0.301029996
8	8 - 17	ACTACAGGATA	0.425968732
9	9 - 18	CTACAGGATAG	0.903089987
10	10 - 19	TACAGGATAGA	0.903089987
11	11 - 20	ACAGGATAGAT	0.726998728
12	12 - 21	CAGGATAGATC	0.726998728
13	13 - 22	AGGATAGATCG	1.329058719
14	14 - 23	GGATAGATCGA	1.204119983
15	15 - 24	GATAGATCGAA	1.028028724
16	16 - 25	ATAGATCGAAT	Infinity
17	17 - 26	TAGATCGAATG	Infinity
18	18 - 27	AGATCGAATGG	Infinity
19	19 - 28	GATCGAATGGT	Infinity
20	20 - 29	ATCGAATGGTG	Infinity
21	21 - 30	TCGAATGGTGG	Infinity

The graphical representation of consecutive len_10 sequence positions and their log_odd_ratio are as shown in Figure 4.

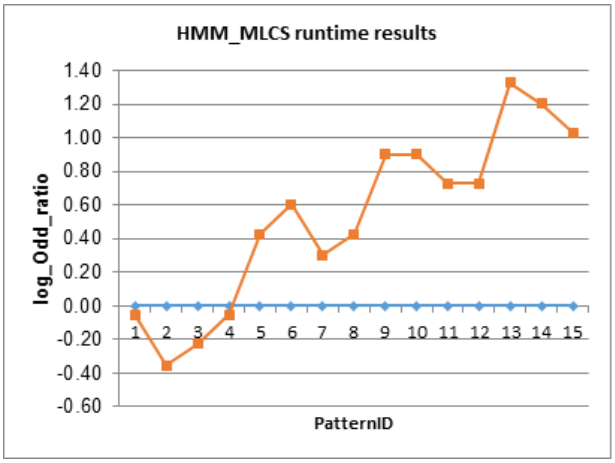

Fig.4.HMM_MLCS runtime results

The runtime results show that log_odd_ratio values for the patternID-5 to patternID-8 is closer to 0.5. i.e. MLCS lies between positions 5 to positions 18 of seq_1. And also Table 7 shows that sequence of len_30 requires only 21 sub-patterns of len_10. And each of the len_10 patterns requires (10 – 1= 9) len_2 patterns to calculate log_odd_ratio values. Thus proposed HMM_MLCS identifies MLCS in linear time and space complexity.

5. CONCLUSION AND FUTURE RESEARCH DIRECTION

The existing DP based algorithms to identify MLCS require exponential space and time complexity. As DNA sequences are of million in length, these algorithms are quite expensive. The statistical method Hidden Markov Model (HMM) is suitable and is proven for the identification of highly aligned sequences. MLCS is also a highly aligned subsequence. HMM based approach identifies MLCS in linear time and space complexity than the score matrices of DP methods.

The proposed algorithm HMM_MLCS defines four states and sixteen len_2 transition states to identify MLCS in linear space and time complexity. In future, HMM_MLCS can be improved by increasing the transition states from sixteen to two fifty six and execute them using Hadoop Map-reduce programming methodology. This approach will further reduce the time and space complexity for the large DNA sequences.

REERENCES

Akutusu T, Dynamic Programming Algorithms for RNA Secondary Structure Prediction with Pseudoknots, Discrete Applied Mathematics, 104, 2000, 45 - 62.

Apostolico A, Browne S, & Guerre C, Fast Linear-Space Computations of Longest Common Subsequences, Theoretical Computer Science, 92, 1992, 3 - 17.

Bentley L, Multidimensional Divide-and-Conquer, ACM, 23(4), 1980, 214 - 229.

Chowdhury A.R, Algorithms and Data Structures for Cache-efficient Computation: Theory and Experimental Evaluation, Univ. of Texas at Austin, PhD Thesis, 2007.

Dayhoff M, Computer Analysis of Protein evolution, Scientific American, 221(1), 1969, 86 - 95.

Fujiwara Y, Asogawa M, & Konagaya A, Stoochastic motif extraction using hidden markov model, Proceedings of the Second International Conference on Intelligent Systems of Molecular Biology, 1994.

Hakata K & Imai H, Algorithms for the Longest Common Subsequence Problem for Multiple Longest Common Subsequence Problem for Multiple Strings Based on Geometric Maxima, Optimization Methods and Software, 10, 1998, 233 - 260.

Hirschberg D, A Linear Space Algorithm for Computing Maximal Common Subsequences, Comm.ACM, 18(6), 1975, 341 - 343.

Hirschberg D, Algorithms for the Longest Common Subsequence Problem, ACM, 24, 1977, 664 - 675.

Kumar S & Rangan C, A Linear Space Algorithm for the LCS Problem, Acta Informatica, 24, 1987, 353 - 362.

Kung H, Luccio F, & Preparata F, On Finding the Maxima of a Set of Vectors, ACM, 22, 1975, 469 - 476.

Le H, & Ramachandran V, Efficient Cache-Oblivious String Algorithms for Bioinformatics, Austin, Dept. of Computer Science, Univ. of Texas at Austin, 2007.

Le H.S, & Ramachandran V, Cache-Oblivious Dynamic Programming for Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 7(3), 2010.

Masek W, & Paterson M, A Faster Algorithm Computing String Edit Distances, Journal of Computer and System Sciences, 20, 1980, 18 - 31.

Rabiner L, & Juang B, An introduction to Hidden Markov Models, IEEE ASSP Magazine, 1986, 4 - 16.

Rick, New Algorithms for the Longest Common Subsequence Problem, Bonn: Computer Science, University of Bonn, 1994.

Sankoff, Matching Seqeunces Under Deletion/Insertion Constaints, Proceedings of of National Academy of Sciences, USA, 1972.

Smith T, & Waterman M, Identification of Common Molecular Subsequences. Journal of Molecular Biology, 147, 1981, 195 - 197.

Stolcke A, & Omohundro S.M, Hidden Markov Model induction by Bayesian model merging, Advances in Neutral Information Processing Systems, 1993, 11 - 18.

Trifonov E, & Berezovsky I, Evolutionary Aspects of Protein Structure and Folding, Current Opition in Structural Biology, 13(1), 2003, 110 - 114.

Vingron M, Near-optimal sequence alignment, Current Opinion in Structural Biology, 1996, 346 - 352.

Wang Q, Korkin D, & Shang Y, A Fast Multiple Longest Common Subsequence (MLCS) Algorithm, IEEE Transactions on Knowledge and Data Engineering, 23(3), 2011.