Shortest Superstring in DNA Sequencing

Rongdeep Pathak
Department of Computer Application
Assam Engineering College
Guwahati, Assam, INDIA

Bichitra Kalita
Department of Computer Application
Assam Engineering College
Guwahati, Assam, INDIA

ABSTRACT
In this paper, a new algorithm has been developed applying graph theoretical approach to study the shortest superstring from a DNA spectrum having variable and fixed length of fragments.

Keywords
DNA sequencing, DNA spectrum, fragments, directed weighted graph, overlap matrix

1. INTRODUCTION
DNA computing is an area where it overlaps computer science, mathematics and molecular biology. In DNA computing, it is generally used biological techniques to efficiently do the computation where the data are represented using the strands of DNA. Computation with the DNA molecule has many advantages over conventional computing methods used in digital computer. In DNA computing, the synthetic DNA molecules are used as information storage and the technique of molecular biology such as gel electrophoresis, enzymatic reactions, polymerase chain reactions etc. are used as computational operations for copying, storing and splitting/ concatenating the information in the DNA molecules respectively. A DNA reaction is much slower than the cycle time of silicon based computer, but it can execute billions of operations simultaneously. The massive parallelism of DNA processing provides the solution of NP complete problems. Therefore DNA computing can be applied to solve large complex computational, optimization, Boolean circuit development, scheduling and many other real life problems such as travelling salesman problem etc.

It was first represented that a DNA polymerase which incorporates an enzyme function for copying DNA is very similar with the function of a Turing machine [1].DNA is a molecule which encodes the genetic information used in the development and functioning of all known living organism and many viruses. DNA molecules are polymers which are built from simple monomer called nucleotides. A nucleotide consists of three components – sugar, phosphate and a base. Nucleotides may differ only by their bases, which are Guanine (G), Adenine (A), Thymine (T) and Cytosine(C). The bases have a pair wise affinity i.e. a pair with T and C pair with G. This is called Waston – Crick complimentarily and it is the central to the formation of double stranded DNA molecule.

Recently many theoretical research works has been done to realize general computation using DNA molecules. A method has been discussed for solving a directed Hamiltonian path problem with seven cities using DNA molecule [1]. Again, it is found that [4] a computational model to realize (via experimental operation of DNA molecule) operations on multiple set of character strings following the encoding of finite alphabet characters into DNA molecules. Further, a new DNA encoding method in which thermodynamic properties of DNA are used to represent numerical values and this method has been applied to solve TSP [8].

In DNA computing the main challenge is encoding data into a DNA sequence. Again in DNA sequencing the main problem is to determine a sequence of nucleotides from an unknown DNA fragment [2, 3]. Generally the input data comes from a biological hybridization experiments which is a set (called spectrum) of words (called fragments/oligonucleotides) over the alphabet {A, C, G, T}. These fragments may have fixed or varying length and usually have overlaps. The main aim is to reconstruct the original DNA sequence of a known length n on the basis of these overlapping words/fragments.

In molecular biology Sequencing By Hybridization (SBH) technique is used in DNA Sequencing. The DNA sequencing problem can be stated as the problem of constructing a string over $\sum$ = {A, C, G, T} from a given spectrum S= { $s_1, s_2, \ldots . s_n$} so that the resulting string is the shortest string which contain as many of the fragments in the spectrum as possible. This problem is called shortest superstring problem.

Several methods for DNA sequencing problems with constant length oligonucleotide have been discussed with the help of Graph theory [7]. An algorithmic approach has been given to find the shortest superstring for ideal spectrum and fragments with constant length by applying the theoretical approach of graph theory [10]. A spectrum without any error is called ideal spectrum. A spectrum may contain positive error that is oligonucleotide present in the spectrum but absent in the original sequence and negative error if oligeonucleotides not present in the spectrum but possible to distinguish in the original sequence. Repetitions of fragments or oligeonucleotides are also treated as negative error.

In this paper, a new algorithm has been discussed to find the shortest superstring from a given spectrum having variable or fixed length of fragments.

This paper is organized as follows – the section 1 focuses some previous works of DNA computing and preliminaries of DNA sequencing. In section 2, a new algorithm has been developed to compute the shortest superstring from a given DNA spectrum. The spectrum may have fixed or varying length of fragments. In section 3, some examples are cited which clarify the algorithm. Section 4 includes the conclusion of this paper.

2. A NEW ALGORITHM TO COMPUTE THE SHORTEST SUPERSTRING FROM A GIVEN SPECTRUM OF FIXED OR VARYING LENGTH OF FRAGMENTS
Here, a directed weighted graph G (V, E, W) has been drawn from a given DNA spectrum. In this spectrum the fragments may have fixed length or varying length.
In this directed weighted graph G (V, E, W), where

V= S, vertex v_i corresponds to fragment s_i

E = {(V_i ,V_j) , if there is a overlap between fragment S_i and fragment S_j where i ≠j

W(V_i ,V_j) = overlap (S_i ,S_j) = $|w_{ij}|$ if i ≠j

Overlap (S_i , S_j) means the longest prefix of S_j that matches a suffix of S_i

For example, let X= ACTGCC and Y= GCCTCAC

ACTGCC

GCCTCAC

Overlap(X, Y) = 3.

After having the directed weighted graph from the given spectrum an overlap matrix M of size (V X V) has been constructed. The entry of the overlap matrix M is as follows-

$$M=A_{ij}=\begin{cases} w(Vi, Vj), \text{where there is an edge from Vi to Vj} \\ 0, \quad \text{if here is no edge between Vi to Vj and } i \neq j \\ \text{NO Value} \qquad\qquad\qquad\qquad \text{if } i = j \end{cases}$$

This overlap matrix M gives the values of overlap between two vertices i.e. the number of overlap between two fragments of the given spectrum.

Now, from this overlap matrix M try to find a path or sequence of vertices, which gives the maximum overlap value. Main objective is to generate sequence of vertices which gives maximum overlap value and should also contain all the vertices of the directed graph. This maximum value sequence is our Shortest Superstring (SS) of the given spectrum.

To compute the Shortest Superstring, an algorithm called Compute_Shortest_Superstring has been used, which gives a set of sequences and the corresponding overlap value.

In the algorithm Compute_Shortest_Superstring, one stack, called stack_of_string, a table, called table_of_complete_sequence and a pointer pointing to the top of the stack_of_string has been used. In the stack_of_string, contains some uncompleted sequences or tours and in the table table_of _complete_ sequence contains complete sequences or tours.

In this approach N, the numbers of ordinary stacks is used where N is the number of vertices i.e the total number of fragments of the given spectrum. Each stack contains values of each rows of the overlap matrix in ascending order without any duplicate value and top of the stack holds the highest value. Also N number of pointers has been used pointing top elements of each stack.

In the algorithm Compute_Shortest_Superstring, the following procedures have been used:

POP () – this procedure remove top element from the stack of string. Returns top element of the stack.

PUSH(X) - insert the string X into the top of the stack of string

Get-last-string(X) - returns the last element of the string X.

Find-max-entry (R_i N) - searches the item N in the i^{th} row of the matrix and if the item is found stores the position in an array. If there is no match then nothing is stored in the array. Returns a pointer to the array.

Find-length(X) – return the size of the array pointing by the pointer X.

Find-max(X) – returns the top element of the X^{th} ordinary stack if the stack is not empty otherwise returns an error value.

COPY(X) – store the string X into the table_of_complete_sequence.

For each vertex, follow the steps of the algorithm Compute_Shortest_Superstring. The initial value of the stack of string is the input vertex and a $ symbol. The input vertex is on the top of the stack. Initially the pointer PTR pointing to the input vertex. The algorithm terminates when the top of the stack is $, i.e. when PTR points to the $ symbols.

ALGORITHM Compute_Shortest_Superstring :
Input: an input vertex

Output: Sequence of vertices

1. String Str;
2. Char LAST;
3. Int Size,N,COUNT=0;
4. Char * M;
5. Str = POP();
6. LAST = Get-last- string(Str);
7. N= find –max(LAST);
8. If (N==Error) then COPY(Str) and go to step 15
9. Else
10. M= find-max-entry(LAST,N);
11. Size=Find-length(M);
12. If (Size==0) then COPY(Str) and Exit;
13. Else if
14. For I = 0 to Size
15. If (Str Π *(M+I) ==Φ
16. Then
17. Str1=Str U *(M+ I)
18. PUSH(Str1);
19. COUNT++;
20. End if
21. End Else if
22. If (COUNT == 0)
23. Then go to step 4
24. End if
25. If (Ptr ≠$) then goto step 2
26. END

The above algorithm gives a set of sequence store in the table_of_complete_ sequence. From these sequences the overlap value for each sequence can be computed. Lets us take $V_i V_j V_K V_l$ is a sequence.

Let, the overlap between V_i and V_j is x, found from the overlap matrix, the overlap between V_j and V_k is y and the

overlap between V_k and V_l is z . Therefore, the total overlap value of the sequence $V_i V_j V_K V_l$ is (x + y + z).

Now, after go through the steps of the above algorithm, a set of sequences and corresponding overlap value has been found. From this information the Shortest Superstring can be computed.

First, separate out the sequences having the maximum value from the table_of_complete_sequence and store these sequences into a new table called Max_Table and follow the following steps:

Case A: (Maximum value sequence having all the vertices or all the fragments of the spectrum)

For each maximum value sequence the following steps has to be followed:

1. Let the maximum value is MAX
2. Find the starting vertex of the sequence
3. Check whether there is a path or overlap from this vertex to the starting vertex of other sequences of table of complete sequence (the starting vertex should not present in the sequence) then adds this starting vertex to the sequence as start vertex.
4. If the total overlap value is new sequence is ≥ MAX then copy this new sequence to the table Max_table

After processing each and every maximum value sequence remove all duplicate sequences and separate out the sequences having highest value and store into a new table called Final_max_table.

The set of sequence or sequence of Final_max_table gives the Shortest Superstring of the given spectrum.

Case B: (Maximum value sequence not contain all the vertices or the fragments of the spectrum)

Case B.1: (Maximum value sequence contain same set of vertices or the fragments of the spectrum)

For each maximum value sequence the following steps has to be followed:

1. Same steps as case A
2. Check is there is any path or overlap from the last vertex to the start vertex then remove the last vertex and placed it at the beginning of the sequence.
3. If the total overlaps value of this new sequence is ≥ MAX then store into Max_table.

After processing each maximum value sequence remove all duplicate sequence and store the highest overlap value sequence into a new table called Max1_table.

Now, separate out all those sequences from the table of complete sequence not containing the vertices of Max1_table.

Same steps follows as above and store the highest overlap value sequence into a table Max2_table.

This process continues until all the vertices are included.

Finally, some tables has been found, like Max1_table, Max2_table and so on. Each table gives shortest substring. By performing computational process the Shortest Superstring of the given spectrum can be computed.

For example, let table Max1_table contains sequence X, Max2_table contains sequence Y and Max3_table contains sequence Z. Sequences X, Y and Z contains all the fragments of the given spectrum. After computational process, the Shortest Superstrings of the given spectrum are as follows:

XYZ, XZY, YXZ, YZX, ZXY and ZYX.

Case B.2: (Maximum value sequence may not contain same set of vertices or the fragments of the spectrum)

Arbitrarily choose any one sequence or set of sequence having same set of vertices or fragments.

Apply same steps as Case B.1

3. JUSTIFY THE ALGORITHM WITH SOME SUITABLE EXAMPLE

Example 1: (Case A)

Suppose the original sequence be s= ACTCTGG and an errorless hybridization experiment generates the ideal spectrum S = {ACT, CTC, CTG, TGG, TCT} [1].

Applying the algorithm, we can reconstruct the original sequence s

Now, draw a directed weighted graph G from the above spectrum S and the vertex set V= {V_1, , V_2, V_3, V_4, V_5}, where

$V_1 = ACT$

$V_2 = CTC$

$V_3 = CTG$

$V_4 = TGG$

$V_5 = TCT$

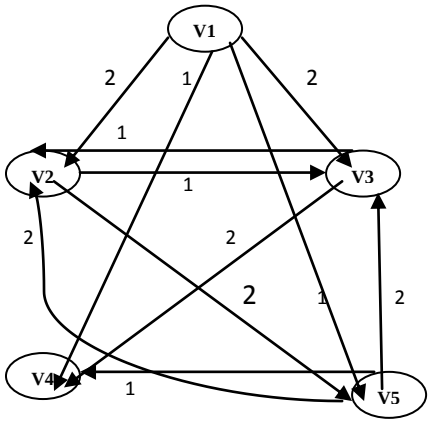

Figure 1: Directed weighted graph for the above spectrum

Table 1: Overlap Matrix

	V1	V2	V3	V4	V5
V1	No_value	2	2	1	1
V2	O	No_value	1	O	2
V3	O	1	No_value	2	O
V4	O	O	O	No_value	O
V5	O	2	2	1	No_value

By applying the algorithm, some set of sequences has been found. The following table 2 is the table_of_complete_sequence, which contains the sequences of the spectrum S and the corresponding overlap value.

Table 2: Table_of_complete_sequence

Sequence	Overlap value
$V_1 V_2 V_5 V_3 V_4$	8
$V_1 V_3 V_4$	4
$V_2 V_5 V_3 V_4$	6
$V_3 V_4$	2
$V_5 V_2 V_3 V_4$	5
$V_5 V_3 V_4$	4

Now, applying the steps of Case A, the sequences that are found, stored in the Max_table. The following table 3 gives the Max_Table and the Final_max _table, which store the shortest superstring of the above spectrum S that is shown in the table 4.

Table 3: Max_table

Sequence	Overlap value
$V_1 V_2 V_5 V_3 V_4$	8
$V_1 V_2 V_5 V_3 V_4$	8

Table 4: Final_max _table

Shortest Superstring
$V_1 V_2 V_5 V_3 V_4$

From the final_max_table, the Shortest Superstring of the above spectrum S = {ACT, CTC, CTG, TGG, TCT} can be computed.

The Shortest Superstring is $s = V_1V_2V_5V_3V_4$ = ACTCTGG

Example 2: (Case A)

Suppose a hybridization experiment generates the spectrum

S= {ACCGT, GTTCGT, CGTGTG, GTGCGT, TCGTG}.

From this spectrum, we have to compute the Shortest Superstring s.

Now, we draw a directed weighted graph G from the above spectrum S and the vertex set V= {V_1, V_2, V_3, V_4, V_5}, where

V_1 = *ACCGT*

V_2 = *GTTCGT*

V_3 = *CGTGTG*

V_4 = *GTGCGT*

V_5 = *TCGTG*

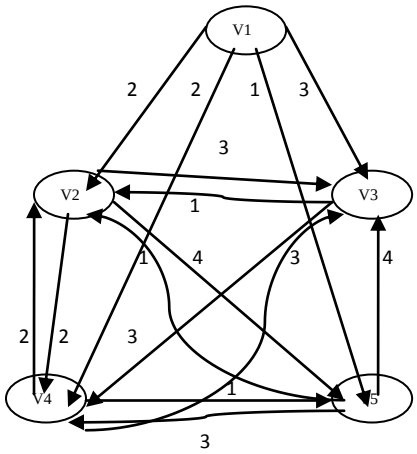

Figure 2: Directed weighted graph for the above spectrum

Figure 2, gives the directed weighted graph of the above spectrum S. Now, from this graph construct the overlap matrix. The following table 5 gives the overlap matrix of the graph G.

Table 5: Overlap Matrix

	V_1	V_2	V_3	V_4	V_5
V_1	No_value	2	3	2	1
V_2	O	No_value	3	2	4
V_3	O	1	No_value	3	O
V_4	O	2	3	No_value	1
V_5	O	1	4	3	No_value

By applying the algorithm, some set of sequences have been found. The following table 6 is the table_of_complete_sequence, which contains the sequences of the spectrum S and the corresponding overlap value.

Table 6: Table_of_complete_sequence

Sequence	Overlap value
$V_1 V_3 V_4 V_2 V_5$	12
$V_2 V_5 V_3 V_4$	11
$V_3 V_4 V_2 V_5$	9
$V_4 V_3 V_2 V_5$	8
$V_5 V_3 V_4 V_2$	9

Now, applying the steps of Case A, some new sequences have been found which are stored in the Max_table. The following table 7 gives the Max_Table and finally, the Final_max _table stores the shortest superstring of the above spectrum S as shown in table 8.

Table 7: Max_Table

Sequence	Overlap value
$V_1 V_3 V_4 V_2 V_5$	12
$V_1 V_2 V_5 V_3 V_4$	13
$V_1 V_3 V_4 V_2 V_5$	12

Table 8: Final_max _table

Shortest Superstring
$V_1 V_2 V_5 V_3 V_4$

From the final_max_table, the Shortest Superstring of the above spectrum S= {ACCGT, GTTCGT, CGTGTG, GTGCGT, TCGTG} has been computed.

The Shortest Superstring is s = $V_1V_2V_5V_3V_4$
= ACCGTTCGTGTGCGT

Example 3 :(Case B)

Suppose, a hybridization experiment generates the spectrum

S= {ACAT, TACA, ATGCAT, TCAAT, CGTGC, GCTACG}

Now,draw a directed weighted graph G from the above spectrum S and the vertex set V= { V_1, V_2, V_3, V_4, V_5 ,V_6},

Where

V_1 = *ACAT*

V_2 = *TACA*

V_3= *ATGCAT*

V_4= *TCAAT*

V_5= *CGTGC*

V_6 = *GCTACG*

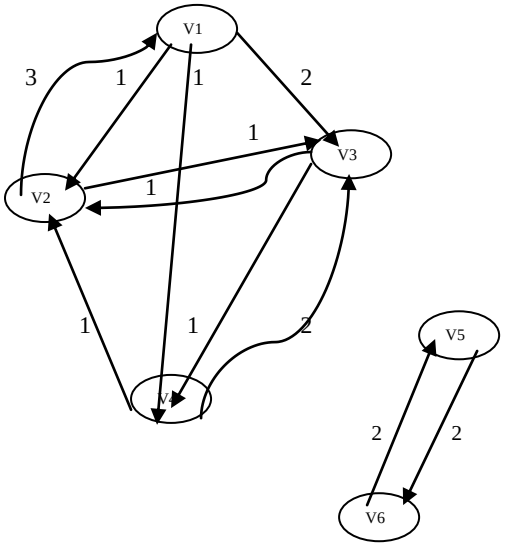

Figure 3: Directed weighted graph for the above spectrum

Figure 3, gives the directed weighted graph of the above spectrum S. Now, from this graph construct the overlap matrix. The following table 9 gives the overlap matrix of the graph G.

Table 9: Overlap Matrix

	V_1	V_2	V_3	V_4	V_5	V_6
V_1	No_value	1	2	1	O	O
V_2	3	No_value	1	O	O	O
V_3	O	1	No_value	1	O	O
V_4	O	1	2	No_value	O	O
V_5	O	O	O	O	No_value	2
V_6	O	O	O	O	2	No_value

By applying the algorithm, some set of sequences have been found. The following table 10 is the table_of_complete_sequence, which contains the sequences of the spectrum S and the corresponding overlap value.

Table10: Table_of_complete_sequence

Sequence	Overlap value
$V_1 V_3 V_2$	3
$V_1 V_3 V_4 V_2$	4
$V_2 V_1 V_3 V_4$	6
$V_3 V_2 V_1 V_4$	5
$V_3 V_4 V_2 V_1$	5
$V_4 V_3 V_2 V_1$	6
$V_5 V_6$	2
$V_6 V_5$	2

Now, applying the steps of Case B.1, the tables Max1_table and Max2_table have been found, which contains all the shortest sub strings as shown in the following Table 11 and Table 12.

Table 11: Max1_table

Sequence	Overlap value
$V_2V_1V_3V_4$	6
$V_4V_3V_2V_1$	6
$V_4V_2V_1V_3$	6

Table 12: Max2_table

Sequence	Overlap value
$V_5 V_6$	2
$V_6 V_5$	2

By performing computational process, the shortest superstring of the above spectrum S can be computed. The following table 13 gives all the shortest Super string of the spectrum S= {ACAT, TACA, ATGCAT, TCAAT, CGTGC, GCTACG}

Table 13: Shortest Superstring

Sequence	Shortest superstring
V_2 V_1 V_3 V_4 V_5V_6	TACATGCATCAATCGTGCTACG
V_2 V_1 V_3 V_4 V_6V_5	TACATGCATCAATGCTACGTGC
V_5V_6 V_2 V_1 V_3 V_4	CGTGCTACGTACATGCATCAAT
V_6V_5 V_2 V_1 V_3 V_4	GCTACGTGCTACATGCATCAAT
V_4 V_3 V_2 V_1 V_5V_6	TCAATGCATACATCGTGCTACG
V_4 V_3 V_2 V_1 V_6V_5	TCAATGCATACATGCTACGTGC
V_5V_6 V_4 V_3 V_2 V_1	CGTGCTACGTCAATGCATACAT
V_6V_5 V_4 V_3 V_2 V_1	GCTACGTGCTCAATGCATACAT
V_4 V_2 V_1 V_3 V_5V_6	TCAATACATGCATCGTGCTACG
V_4 V_2 V_1 V_3 V_6V_5	TCAATACATGCATGCTACGTGC
V_5V_6 V_4 V_2 V_1 V_3	CGTGCTACGTCAATACATGCAT
V_6V_5 V_4 V_2 V_1 V_3	GCTACGTGCTCAATACATGCAT

4. CONCLUSION

This paper mainly focuses a new algorithm to find the shortest superstring of a given DNA spectrum having variable or fixed length of fragments. In this algorithm the graph theoretical approach has been used. This algorithm gives all the possible shortest superstrings that may be present in a given DNA spectrum of fixed or varying length of fragments.

5. REFERENCES

[1] Adleman LM, Molecular computation of solution to combinatorial problems, Science, 266: 1021-1024, 1994

[2] MS Wateman, Introduction to computational biology, Maps, sequences and Genomes, Chapman & Hall , London, 1995.

[3] J Setubal and J Meidanis, Introduction to computational molecular biology, PWS Publishing Company, Boston, 1997.

[4] Adleman LM, Computing with DNA, Scientific American, 29(2), 54-61, 1998.

[5] PA Pevzner, Computational Molecular biology, an algorithmic approach, MIT Press, Cambridge, 2000.

[6] J Blazewicz and M Kasprzak, Complexity of DNA sequencing by hybridization , Theatrical computer science 290, 1459-1473, 2003.

[7] M Kasprzak, On the link between DNA sequencing and graph theory, Computational Methods in Science and Technolgy, 10, 39-47, 2004.

[8] JY Lee, SY Shin, TH Park, BT Zhng, Solving travelling salesman problems with DNA molecule encoding numerical values, Biosystems, Elsevier, 2004

[9] K Mehdizadeh, MA Nekoui, K Sabahi and A Akbarimajd, A modified DNA computing algorithm to solve TSP, IEEE, 2006.

[10] P Kalita and B Kalita, A graph theoretical algorithmic approach for DNA sequencing, IOSR, Vol. 5 issue 1, 40-46, 2013.