TABLE OF CONTENTS

PROBLEM STATEMENT

STATEMENT-1

Given a set of m DNA sequences each of length n – nucleotides ,find the pattern p of length l that is repeated in all the sequences.

STATEMENT-2

Givan a set of l-mers ,construct the set of most probable consensus strings (motif) through profiling.

INTRODUCTION

Many immunity genes in the genome have strings that are reminiscent of TCGGGGATTTCC, located upstream of the genes' start. These short strings, called *NF-κB binding sites*, are examples of *regulatory motifs*. These regulatory motifs turn on immunity and other genes.

Proteins known as *transcription factors* bind to these motifs, encouraging RNA polymers to transcribe the downstream genes.

Motif finding is the problem of discovering such motifs without any prior knowledge of how the motif looks.

Seven random sequences

CGGGGCTGGGTCGTCACATTCCCCTTTCGATA
TTTGAGGGTGCCCAATAACCAAAGCGGACAAA
GGGATGCCGTTTGACGACCTAAATCAACGGCC
AAGGCCAGGAGCGCCTTTGCTGGTTCTACCTG
AATTTTCTAAAAAGATTATAATGTCGGTCCTC
CTGCTGTACAACTGAGATCATGCTGCTTCAAC
TACATGATCTTTTGTGGATGAGGGAATGATGC

Figure presents seven 32-nucleotide DNA sequences generated randomly. Also shown in below figure are the same sequences with the "hidden" pattern

P = ATGCAAC of length l = 7 implanted at random positions.

The same DNA sequences with the implanted pattern ATGCAAC

CGGGGCTATGCAACTGGGTCGTCACATTCCCCTTTCGATA
TTTGAGGGTGCCCAATAAATGCAACTCCAAAGCGGACAAA
GGATGCAACTGATGCCGTTTGACGACCTAAATCAACGGCC
AAGGATGCAACTCCAGGAGCGCCTTTGCTGGTTCTACCTG
AATTTTCTAAAAAGATTATAATGTCGGTCCATGCAACTTC
CTGCTGTACAACTGAGATCATGCTGCATGCAACTTTCAAC
TACATGATCTTTTGATGCAACTTGGATGAGGGAATGATGC

Suppose you do not know what the pattern P is, or where in each sequence it has been implanted. We have to reconstruct P by analyzing the DNA sequences.We could simply count the number of times each l-mer, or string of length l, occurs in the sample. Since there are only $7 \cdot (32 + 8) = 280$ nucleotides in the sample, it is unlikely that any 8-mer other than the implanted pattern appears more than once.9 After counting all 8-mer occurrences we will observe that, although most 8-mers appear in the sample just once (with a few appearing twice), there is one 8-mer that appears in the sample suspiciously many times—seven or more. This overrepresented 8-mer is the pattern P we are trying to find.

DATA STRUCTURES USED

ARRAYS-An **array**, is a **data structure** consisting of a collection of elements (values or variables), each identified by at least one **array** index or key. An **array** is stored so that the position of each element can be computed from its index tuple by a mathematical formula.

TERNARY SEARCH TREE is a type of tree (sometimes called a *prefix tree*) where nodes are arranged in a manner similar to a binary search tree, but with up to three children rather than the binary tree's limit of two. Like other prefix trees, a ternary search tree can be used as an associative map structure with the ability for incremental string search. However, ternary search trees are more space efficient compared to standard prefix trees, at the cost of speed. Common applications for ternary search trees include spell-checking and auto-completion.

DETAILED DESIGN OF THE PROJECT

Superposition of the seven highlighted 8-mers

CGGGGCTATcCAgCTGGGTCGTCACATTCCCCTT...
TTTGAGGGTGCCCAATAAggGCAACTCCAAAGCGGACAAA
GGATGgAtCTGATGCCGTTTGACGACCTA...
AAGGAaGCAACcCCAGGAGCGCCTTTGCTGG...
AATTTTCTAAAAAGATTATAATGTCGGTCCtTGgAACTTC
CTGCTGTACAACTGAGATCATGCTGCATGCcAtTTTCAAC
TACATGATCTTTTGATGgcACTTGGATGAGGGAATGATGC

A T C C A G C T

G G G C A A C T

A T G G A T C T

ALIGNMENT A A G C A A C C

MATRIX T T G G A A C T

A T G C C A T T

A T G G C A C T

	A	5 1 0 0 5 5 0 0
PROFILE	**T**	1 5 0 0 0 1 1 6
MATRIX	**G**	1 1 6 3 0 1 0 0
	C	0 0 1 4 2 0 6 1

CONSENSUS A T G C A A C T

The alignment matrix, profile matrix and consensus string formed from the 7-mers starting at positions

s = (8,19,3,5,31,27,15) as shown above.

From above example, to alignment matrix, to profile, and, finally, to consensus string. If s = (8,19,3,5,31,27,15) is an array of starting positions for 8-mers, then Score(s) = 5 + 5 + 6 + 4 + 5 + 5 + 6 + 6 = 42.

Each node in a ternary search tree contains only 3 pointers-
1. The left pointer points to the node whose value is less than the value in the current node.
2. The equal pointer points to the node whose value is equal to the value in the current node.
3. The right pointer points to the node whose value is greater than the value in the current node.

Apart from above three pointers, each node has a field to indicate data(character in case of dictionary) and another field to mark end of a string.

IMPLEMENTATION DETAILS AND RESULTS

We have implemented these problem statements using arrays and ternery search tree and were have to find our motif.

CONCLUSION

Hence we were able to solve both the problem statements using arrays and ternary search tree and were able to find our motif.