A

Mini project Report

On

" GENE FOLDING"

Submitted By

Sr.no	Name	Roll no
1.	Atharv Prakash Sangar (GL)	40
2.	Anuj Digambar Kadam	28
3.	Avdhut Shankar Dhumal	42
4.	Atharv Jagannath patil	35
5.	Asharf Riyaz Hukkerikar	34

Under the Guidance of

Dr .VIDYA BADADARE

Department of Computer Science & Engineering

School of Engineering & Technology

D. Y. Patil Agriculture & Technical University, Talsande

Academic Year: 2023-24

D. Y. Patil Agriculture & Technical University, Talsande

Department of Computer Science & Engineering

Certificate

This is to certify that the Mini Project work entitled

"GENE FOLDING"

submitted by

NAME	PRN NO
1. Atharv Prakash Sangar	2022011031197
2. Anuj Digambar Kadam	2022011031071
3. Avdhut Shankar Dhumal	2022011031040
4. Atharv Jagannath Patil	2022011031139
5. Asharf Riyaz Hukkerikar	2022011031058

In partial fulfillment of requirement for the Semester-I of Second Year in Computer Science & Engineering. This is a record of their work carried out by them under supervision and guidance during academic year 2023-24.

Place: DYP ATU TALSANDE
Date: 15 /12/2023

Mrs.Vidya Badadare	Mr.S.A.Kumbhar	Dr.S.T.Patil	
Guide	**Project coordinator**	**HOD CSE**	**External Examinor**

ACKNOWLEDGMENT

We take pleasure in presenting our work done for the project entitled as "Gene folding".

We express our sincere thanks to **DR .Vidya Bhadadare** whose supervision, inspiration and valuable guidance helped us a lot to complete the project. Her guidance proved to be the most valuable to overcome all the hurdles in the fulfillment of this project work.

We are very much thankful to **Mr. S. A. Kumbhar**, Project coordinator and **Dr. S. T. Patil, HOD CSE** for their kind support and valuable guidance.

We would also express gratitude towards our colleagues and friends for the moral and technical support throughout the duration of project work. Also we are thankful to all those who have helped us in the completion of the project work.

Atharv Prakash Sangar(GL)

Anuj Digambar Kadam

Avdhut Shankar Dhumal

Atharv Jagannath Patil

Asharf Riyaz Hukkerikar

[Note: sponsorship Letter and progress report should be added after this page]

CHAPTER:1
INTRODUCTION

1.Introduction

1.1 Problem Statement

International Cell Processing Company (ICPC) is a world leader in the analysis of genetic sequences. A genetic sequence is a sequence of nucleotides, which in this problem is represented by a string containing only the letters A, C, G, and T in some combination, each letter representing a single nucleotide (Adenine, Cytosine, Guanine, and Thymine, respectively).

One of the key discoveries made by ICPC is that through a process called Genetically Optimized Organic Folding (GOOF), they can take a genetic sequence and transform it into a simpler one, while preserving many of the properties of the sequence that ICPC wants to analyze.

A single application of GOOF works as follows. Find a point between two adjacent nucleotides in the nucleotide sequence, such that the sequence reads the same from that point in both directions, up until the nearer end of the sequence. For instance, in the sequence ATTACC, there are two such points: AT-TACC and ATTAC-C. Then pick one of these points (say, the first one), and fold the genetic sequence at that point, merging the identical nucleotides (so, in this case the AT and TA would become merged, and the resulting sequence would be CCAT or TACC).

Through repeated application of GOOF, a nucleotide can potentially be made much shorter. However, manually searching for the appropriate folding points is very time-consuming. ICPC reached out to you to write a program that would automate the process of finding the folding points and choosing them so as to obtain the shortest possible genetic sequence from a given input sequence.

1.2 Problem Description :

The International Cell Processing Company (ICPC) has pioneered a ground breaking technique known as Genetically Optimized Organic Folding (GOOF) for simplifying genetic sequences. A genetic sequence is represented as a string composed of the nucleotides A, C, G, and T, denoting Adenine, Cytosine, Guanine, and Thymine, respectively.

GOOF involves iteratively identifying specific folding points within a genetic sequence, where folding results in a simplified sequence while retaining key properties for analysis. The process is as follows:

point between two adjacent nucleotides in the sequence where, when the sequence is read from that point in both directions, it remains identical until reaching the nearer end of the sequence.

Choose one of these identified folding points and fold the genetic sequence at that point, merging identical nucleotides.

The objective is to automate the application of GOOF to find folding points and select them strategically to obtain the shortest possible genetic sequence from the given input sequence. Manual searching for optimal folding points is time-consuming, prompting ICPC to seek a program that efficiently performs this task.

Your task is to develop a program that automates the process of applying GOOF to genetic sequences, streamlining the identification and selection of folding points to achieve the shortest possible sequence length.

1.3 Software Requirement Specification :

Abstract :

Company Overview: International Cell Processing Company (ICPC) is a global leader in genetic sequence analysis.

Genetic Sequences: Genetic sequences are represented by strings of nucleotides (A, C, G, T) and are subject to analysis by ICPC.

Key Discovery - GOOF: ICPC employs Genetically Optimized Organic Folding (GOOF) to simplify genetic sequences while preserving essential properties.

GOOF Process:

Identify a point between two adjacent nucleotides where the sequence reads the same in both directions.

Choose one of these points and fold the sequence, merging identical nucleotides.

Example: In the sequence ATTACC, two folding points are AT-TACC and ATTAC-C. Choosing one, like AT-TACC, results in a folded sequence such as CCAT or TACC.

Introduction :

Purpose :

This document outline the requirement for the development of a software program to automate the process of finding folding points and selecting them to obtain the shortest possible genetic sequence using genetically optimized organic folding (GOOF) as requested by the International cell processing company (ICPC)

The purpose of this Software Requirements Specification (SRS) is to provide a comprehensive overview of the requirements and specifications for developing

a C-based program to solve the gene folding problem. This document aims to define the goals, functionalities, and constraints of the software system.

Scope :

The software will analysed genetic sequences composed of the letters A,C,G and T to determine optimal folding points and perform the folding points and perform the folding process to minimize sequence length while preserving essential properties for analysis

References :

ACM-ICPC 2020 Problem set: D

https://en.m.wikipedia.org/wiki/Protein_folding

https://comis.med.uvm.edu/VIC/coursefiles/MD540/MD540-Protein_Organization_10400_574581210/Protein-org/Protein_Organization8.html

Developer's Responsibilities :

The developer is responsible for :

(a) Developing the system.
(b) Installing the software on the client's hardware.
(c) Conducting any user training that might be needed for using the system.
(d) Implement the requirements.
(e) Testing.
(f) Performance Optimization.
(g) Maintaining the system for the period of one year after installation.

General Description :

Product Function Overview :

Product Functions:

- Input Gene Sequences

a. Allow users to input gene sequences in a specified format.

b. Validate the input gene sequences for correctness and adherence to the required format.

c. Provide error messages or prompts for incorrect or invalid input.

Algorithm Execution:

a. Implement an algorithm to predict the folding patterns of gene sequences.

b. Execute the algorithm on the input gene sequence

c. Optimize the algorithm for efficient execution and minimize computational time.

User Characteristics :

The users for the Visual Python++ parsing problem are programmers , instructors and educators with knowledge of syntax rules and rectangles. They seek to efficiently develop a feature that matches corner pairs to create nested rectangles without overlap. Their characterstics include logical thinking and attention to detail.

Genetic Sequence Input:

The software shall allow the user to input a genetic sequence composed of letters 'A', 'C', 'G', and 'T'.

Find Folding Points:

The software shall analyse the input genetic sequence to identify valid foldingg points.
A valid folding point is a location in the sequence where it reads the same from that point in both directions.

Fold Sequence:

The software shall fold the genetic sequence at the identified folding points, merging identical nucleotides to create folded sequences.

Optimize Sequence:

The software shall choose the shortest folded sequence from the list of folded sequences generated in the previous step.

The selected sequence shall be the result of the optimization process.

output generation:-

Provide options for generating output files or reports containing the folding predictions.

Include relevant information and metadata in the output files, such as gene sequence details and prediction confidence levels.

Constraints:

The program can run on the device which supports on C/C++ compilers.

System Interfaces:

The software will have a graphical user interface (GUI) for user interaction. It may use standard libraries and frameworks for string manipulation and user interface development.

System Testing:

The software shall undergo testing to ensure that it correctly identifies folding points and optimizes genetic sequences.

Future Enhancements

In the future the system could incorporate additional algorithms for optimizing the folding process.

Integration with database and analytical tool for genetic analysis.

Input and Outputs :

Input :

The input contains a single string s representing the nucleotide sequence to be analyse. The string

consists of characters A, C, G, and T only. The length of s is between 1 and 4 · 10 enclusive

Output :

Output the smallest possible length of a sequence obtained from the input by applying GOOF zero or more times.

Sample input 1 **sample output 1**

ATTACC	**3**

Sample input 2 **sample output 2**

AAAAGAATTAA	**5**

Input Sequence :--

The Initial Genetic Sequence Provided To The System

Sequence Analysis:--

Analyzing The Genetic Sequence To Identify Potential Folding Points.

Find Folding Points :--

Determining Points In The Sequence Where Folding Is Possible

Choose Folding Points:--

Selecting An Optional Folding Points Among The Identified Options

Apply GOOF Process:--

Executing The Genetically Optimized Organic Folding Process At The Chosen Points

Output Shortened Sequence :--

The Resulting Shortened Genetics Sequence After Applying The GOOF Process

Modules:--

Input Module:--

Responsible for taking the initial genetic sequence as input and obtain information of genetic acides and pass it through next leveland pass it to the super ordinate

Analysis Module:--

Submodule 1: Identify Folding Points

Analyzes the genetic sequence to find points where folding can occur.

Submodule 2: Optimal Folding Point Selection

Evaluates potential folding points to determine the one that results in

the shortest genetic sequence.

Transform module transform the data from one to other form

Displays or outputs the final, simplified genetic sequence obtained

CHAPTER : 2

DESIGN

2.DESIGN DOCUMENT

Function Oriented Design Is Done By The Following Steps:--

Step 1:-restate the problem as a data flow diagram

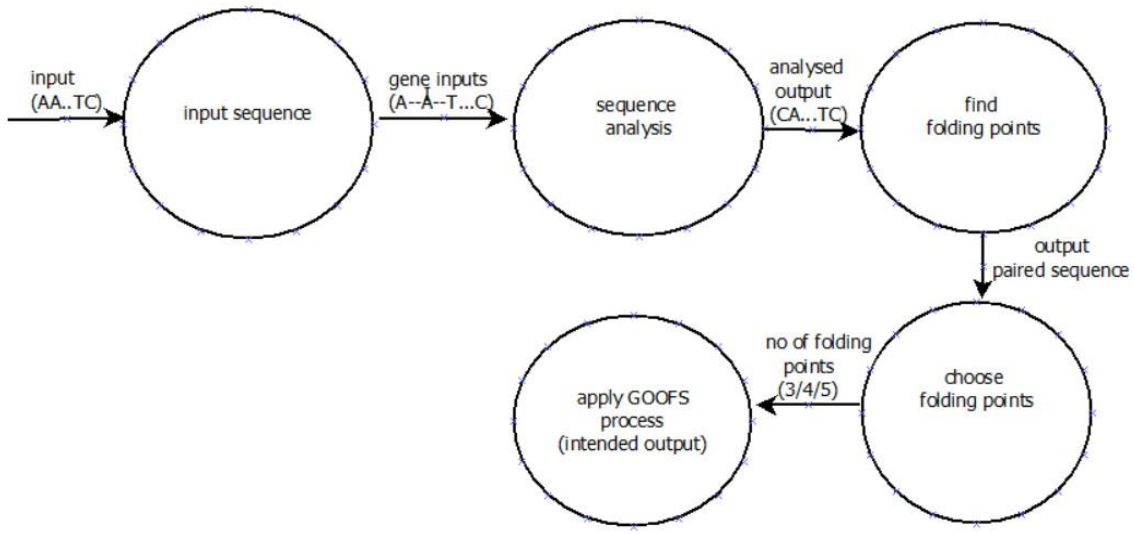


Fig a.Dataflow diagram level 0

The Data Flow Diagram Is Representing The Flow Of Data As Following :

Input Sequence :-- The Initial Genetic Sequence Provided To The System

Sequence Analysis:-- Analyzing The Genetic Sequence To Identify Potential Folding Points.

Find Folding Points :-- Determining Points In The Sequence Where Folding Is Possible

Choose Folding Points:-- Selecting An Optional Folding Points Among The Identified Options

Apply GOOF Process:-- Executing The Genetically Optimized Organic Folding Process At The Chosen Points

Output Shortened Sequence :-- The Resulting Shortened Genetics Sequence After Applying The GOOF Process

Step 2:-- Identify the most abstract input and output :--

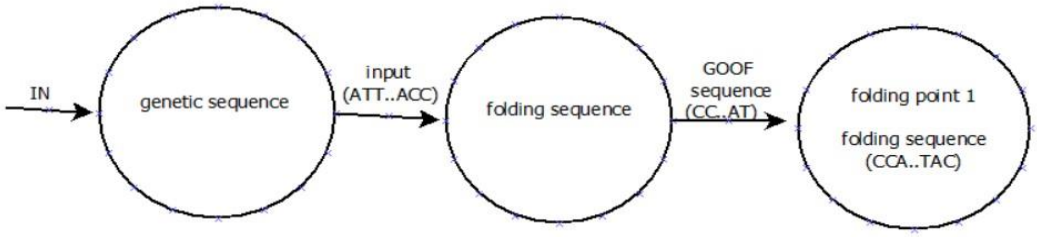


Fig b.Most abstract input and output

step 3:-- structure chart

The structure chart is graphical representation of the system. A chart shows the breakdown of a system to its lowest manageable levels. This structure chart includes 3 different types of modules:

Input Module:--
Responsible for taking the initial genetic sequence as input.and obtain information of genetic acides and pass it through next level and pass it to the super ordinate

Analysis Module:--

Submodule 1: Identify Folding Points

Analyzes the genetic sequence to find points where folding can occur.

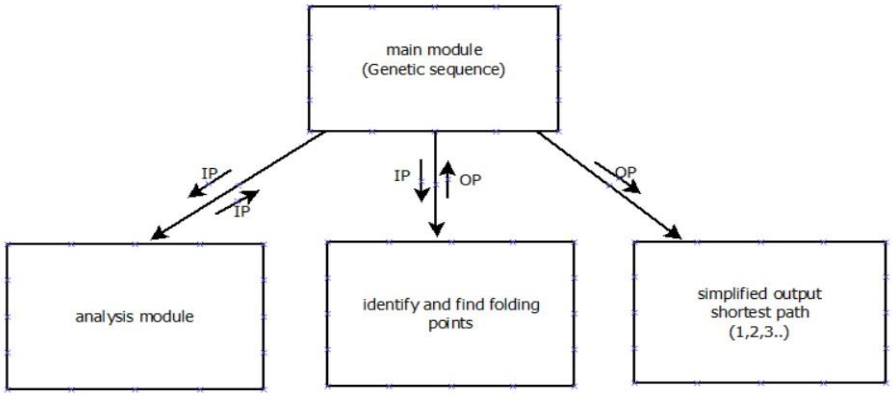


Fig .c Structure chart

Submodule 2: Optimal Folding Point Selection

Evaluates potential folding points to determine the one that results in the shortest genetic sequence.

Transform module transform the data from one to other form

Output Module:--

Displays or outputs the final, simplified genetic sequence obtained through the Genetically Optimized Organic Folding process

CHAPTER:3

CODING

3.CODING

```
#include <stdio.h>
#include <string.h>
#define MAX_SEQUENCE_LENGTH 4000001
// Function to find the folding point and update the genetic sequence
void applyGOOF(char sequence[]) {
 int length = strlen(sequence);
 // Iterate through each possible folding point
 for (int i = 1; i < length; ++i) {
 int left = i - 1, right = i;
 // Check for identical nucleotides on both sides
 while (left >= 0 && right < length && sequence[left] == sequence[right]) {
 --left;
 ++right;
 }
 // If a folding point is found, update the sequence
 if (right - left - 1 > 1) {
 // Merge identical nucleotides
 memmove(&sequence[left + 2], &sequence[right], length - right + 1);
 length -= right - left - 1;
 // Restart the iteration to find more folding points
```

```
 i = 0;
 }
 }
}
int main() {
 char sequence[MAX_SEQUENCE_LENGTH];
 // Input nucleotide sequence
 printf("Enter the nucleotide sequence: ");
 fgets(sequence, sizeof(sequence), stdin);
 // Remove newline character from input
 size_t len = strlen(sequence);
 if (len > 0 && sequence[len - 1] == '\n') {
 sequence[len - 1] = '\0';
 }
 // Apply GOOF process
 applyGOOF(sequence);
 // Output the length of the sequence
 printf("Length of the sequence: %zu\n", strlen(sequence));
 return 0;
}
```

CHAPTER:4

TESTING

4.TEST CASE DOCUMENT

1.

Test Case ID	TC001	**Test Case Description**	Test the Sum Module		
Created By	Anuj Kadam	**Reviewed By**	Avdhut Dhumal	**Version**	2.1
Tester's Name	Atharv patil	**Date Tested**	December 6, 2023	**Test Case (Pass/Fail/Not**	Pass

TCNO.	Input Data	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	AAAAATTTCCC	10	10	PASS
2	AAAATTCC	7	7	PASS
3	TTCCGGA	6	5	FAIL
4	CCTTTTTAA	8	8	PASS

2.

Test Case ID	TC002	Test Case Description			
Created By	Anuj kadam	Reviewed By	Avdhut dhumal	Version	2.2

QA Tester's Log

Tester's Name	MR.S.A.Kumbhar	Date Tested	December 7, 2023	Test Case (Pass/Fail/Not	PASS

S #	Prerequisites:
1	Valding folding points
2	expected output
3	edge class
4	randomizing sequence

S #	Test Data Requirement
1	genetic sequence
2	genetic sequence
3	genetic sequence
4	genetic sequence

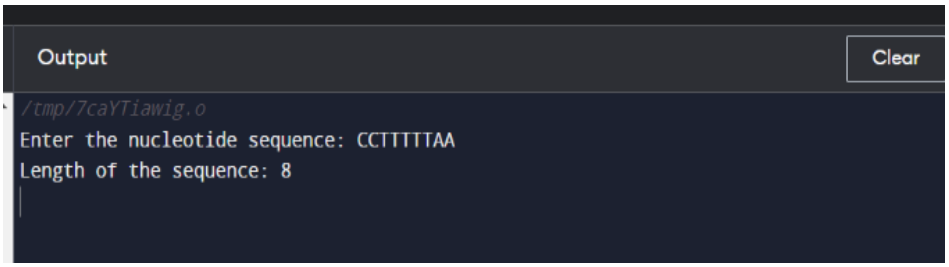
Test Conditions

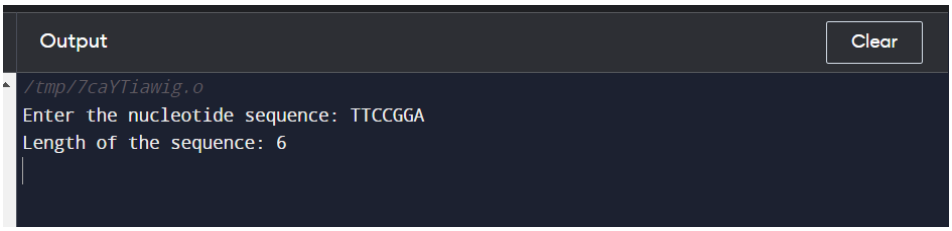
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	empty string	no error	no error	PASS
2	AAAAGAATTAA	5	5	PASS
3	TTCCGGA	6	6	PASS
4	ATTACC	5	5	PASS

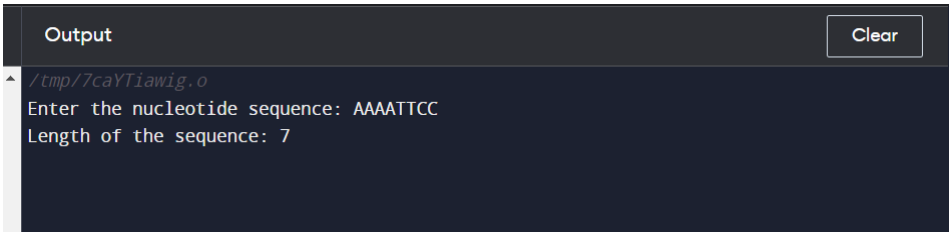
CHAPTER : 5

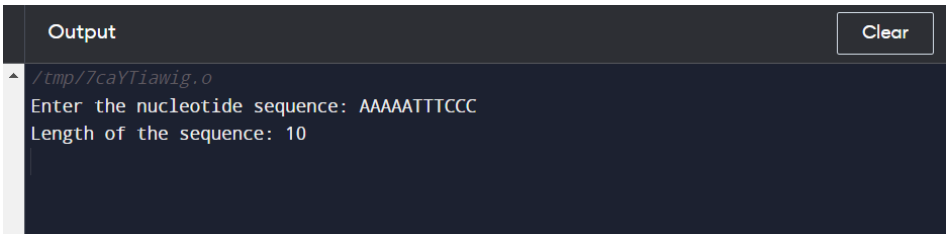
RESULT

SCREENSHOTS:--









6.REFERENCE AND CONCLUSION

Reference:-

ACM-ICP

https://en.m.wikipedia.org/wiki/Protein_folding

https://www.news-medical.net/life-sciences/Protein-Folding.aspx

conclusion:--

This software requirement specification outlines the requirements for the development of a software program that automates the process of finding folding points and choosing them to obtain the shortest possible genetic sequence from a given input sequence using the GOOF process as requested By the international cell processing company (ICPC)the software aims to streamline the analysis of genetics sequence for research and analysis purposes