



## **Enhancement of Twin Gene System**

**By:**

Ruba Yousef Sultan

Ruba Diab Al Hirbawe

Manar Wadah Al Hammoury

**Supervisor:**

Dr Mahmoud Al Saheb

Dr Yaqoub Ashhab

### **Software Project Report**

Submitted to the Department of Electrical and Computer Engineering in the college of  
Engineering and Technology

**Palestine Polytechnic University**

This project is submitted in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Systems Engineering

**Palestine Polytechnic University**

**Hebron – Palestine**

Jan 2005



## **Enhancement of Twin Gene System**

**By:**

Ruba Yousef Sultan

Ruba Diab Al Hirbawe

Manar Wadah Al Hammoury

**Supervisor:**

Dr Mahmoud Al Saheb

Dr Yaqoub Ashhab

### **Software Project Report**

Submitted to the Department of Electrical and Computer Engineering in the college of  
Engineering and Technology

**Palestine Polytechnic University**

This project is submitted in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Systems Engineering

**Palestine Polytechnic University**

**Hebron – Palestine**

Jan 2005

# **Enhancement of Twin Gene System**

**By:**

Ruba Yousef Sultan

Ruba Diab Al Hirbawe

Manar Wadah Al Hammoury

**Supervisor:**

Dr Mahmoud Al Saheb

Dr Yaqoub Ashhab

## **Software Project Report**

This project is submitted in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Systems Engineering

**Palestine Polytechnic University**

2004/2005

Approved by \_\_\_\_\_  
Chairperson of Supervisory Committee

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Program Authorized  
to Offer Degree \_\_\_\_\_

Date \_\_\_\_\_

## ABSTRACT

This project is an enhancement of the existing "Twin Gene" system which was introduced last year as a graduation project

This system strives to test the old system and determine the aspects of modification, improve the interface of the old twin gene system, improve the system performance and add new capabilities to the system.

Improving an existing old system is important to all biologists, because they need to perform all their functions in more efficient way, and inside friendly user interface.

المشروع هو تعديل على مشروع سابق بعنوان "الجين التوأم" قُدم السنة الماضية كأحد متطلبات التخرج. من أهداف المشروع اختبار النظام السابق لتحديد مواطن التعديل, وتحسين واجهات البرنامج لتسهيل تفاعل م مع النظام, بالإضافة إلى تحسين أداء النظام و زيادة وظائف جديدة له.

تطوير نظام في مجال الجينات مهم للبيولوجيين حتى يساعدهم في أداء عملياتهم بفعالية أكبر من خلال بيئة تفاعلية سهلة, و لمواكبة التطور في علم الجينات.

## **DEDICATION**

To our Martyrs for their scarification

To our families for their patience

To our brothers and sisters

To our colleagues for their support and encouragement

To our Supervisor Dr. Mahmoud Al Saheb for his ideas, supports and advices.

## ACKNOWLEDGMENT

They said, “Who did not thank people, will not thank God”.

Authors would like to thank our supervisor Dr.Mahmoud Al-Saheb and Dr.Yaqoub Al-Ashhab for their helpful supervising and directions.

## TABLE OF CONTENTS

Abstract.....	IV
Dedication .....	V
Acknowledgement.....	VI
Table of Contents .....	VII
List of Tables.....	X
List of Figures.....	XI
Chapter One	
Introduction .....	1
1.1 Overview.....	1
1.2 Project Importance.....	3
1.3 Project Objectives .....	3
1.4 The Current System.....	4
1.5 The Report Outline .....	5
Chapter Two	
System Planning .....	7
2.1 Overview .....	7
2.2 System Risks .....	7
2.3 Development Requirement .....	7
2.3.1 Hardware Development Requirement .....	7
2.3.2 Software Development Requirement .....	8
2.3.3 Human Development Requirement .....	8
2.4 Cost Estimation .....	9
2.4.1 Development Cost .....	9
2.5 Operational Requirement .....	10
2.6 Time Scheduling .....	10
Chapter Three	
System Requirement .....	11
3.1 Overview .....	11

3.2 Main Requirement .....	11
3.3 Functional Requirement .....	13
3.3.1 Functional Requirement Definition .....	13
3.3.2 Functional Requirement Specification .....	14
3.4 Non Functional Requirement .....	28
Chapter Four	
Initial Testing .....	30
4.1 Overview .....	30
4.2 User Interface Testing .....	30
4.3 Processes Testing .....	31
Chapter Five	
System Analysis and Design .....	34
5.1 Overview .....	34
5.2 Data Design .....	34
5.2.1 Data Structural Design .....	34
5.3 Data needed for the system .....	38
5.3.1 System Files .....	38
5.3.2 System Parameters .....	38
5.4 Design Models .....	39
5.4.1 System Block Diagrams .....	39
5.4.2 Data Flow Diagram .....	40
5.4.3 System Structure Chart .....	41
5.4.4 System Flow Charts .....	42
5.5 Interface Design .....	54
5.5.1 Main Window .....	54
5.5.2 Multiple Alignment Window .....	57
Chapter Six	
Implementation .....	63
6.1 Overview .....	63
6.2 Building The System .....	63
6.3 Prepare the platform of the system .....	63



6.3.1 Install and configure windows XP professional .....	63
6.3.2 Install Visual Basic.Net .....	64
6.3.3 Install BLAST Program.....	64
6.4 Multiple Alignment Implementation .....	65
6.5 Single Alignment Implementation .....	68
6.6 Global Alignment Implementation .....	71
6.7 Grouping Implementation .....	74
Chapter Seven	
System Testing .....	80
7.1 Overview .....	80
7.2 Testing Procedures .....	80
7.2.1 Unit Testing .....	80
7.2.2 Module Testing .....	81
7.2.3 Subsystem Testing .....	83
7.2.4 System Testing .....	83
Chapter Eight	
Deployment and Maintenance .....	84
8.1 Overview.....	84
8.2 Establishment of production Environment.....	84
8.3 Migration and Deployment Plan.....	84
8.4 Implementation Plan.....	85
8.5 Installing Twin Gene System.....	85
8.6 Maintenance Plan .....	86
Chapter Nine	
Conclusion, Problems and Future Work .....	90
9.1 Conclusion .....	90
9.2 Problems .....	90
9.3 Future Work .....	91
References.....	92

## **LIST OF TABLES**

2.1 Hardware Development Resources .....	7
2.2 Software Development Resources .....	8
2.3 Development Hardware Cost .....	9
2.4 Development Software Cost .....	9
2.5 Development Human Cost .....	9

## LIST OF FIGURES

2.1 Time Scheduling .....	10
5.1 System Block Diagram .....	39
5.2 Data Flow Diagram .....	40
5.3 System Structure Chart .....	41

# Chapter One

## Introduction

### 1.1 Overview

The world inclination today mainly focuses on mixing technologies and deploying them to standardize the process of solving problems.

Computer science is related to all sciences and to all aspects of our life; its relation to biology science is strengthened by Bioinformatics which is defined as conceptualizing biology in terms of molecules and then applying informatics techniques to understand and organize information associated with these molecules.

No way to escape from new technologies. We are sensing and living in some new standards of living. The computer science besides biology science help in developing very intelligent systems .

This project is an enhancement of the existing “Twin Gene” system which was introduced last year as a graduation project by Gharam AlSkafi, Ala Abu Al-Reesh and Nida Al-Amleh .

Twin gene system can be thought as a system that allows the user to make some processing on a required gene to find its similar genes, determining the groups of the genes contained in one database of genes.

The importance of such systems make the computer scientists and engineers focus their sights to solve the problems facing the biologists, because all the world inclination is to do all the work in fast and easy ways, so the computer solves this very efficiently .

The old Twin Gene system solves the problems that faces the biologist during manual processing for genes, but some processes does not work properly and others are time consuming .

The new system tries to solve all the problems found in the old twin gene system in order to match the user requirement. It will be introduced to test the previous system and validate and improve it .It will also add new processes that the biologist couldn't do before:

- Detect and recover errors.
- Improve the system interface.
- Improve the system performance.
- Correct the grouping algorithm, and create the new gene that resulted from the different groups, and change the way of displaying its results.
- Create a process that translate the character representation of the genes to proteins.
- Add a process that compose a twin gene from the similar genes resulted from grouping.
- Make enhancement on the global alignment, and making it according to the score value of each EST(Expressed Sequence Tag) that produced by blast output, and displaying it using graphics instead of displaying it on external HTML file.
- Enable the user to import BLAST results from Internet , and apply it to this system.
- Simplify the way of importing the query gene .
- Enable the user to view the content of blast result that related to each EST gene.
- Allow the user to view the results of multiple sequence alignment on the same screen, instead of displaying it on external HTML file .
- Make the program portable that can be executed under any platform .

In addition, the system will enable the biologist to perform some new processes, that make the work with genes easier than before, also the new system will solve the problem of system validation messages, and some errors found in the old system in order to deploy it in biological centers. This will be implemented via a development environment called Visual Basic.NET, because this environment helps providing good tools for improving the interface, and because this is the environment that the original system is built on.

## 1.2 Project Importance

As the number of the projects that deal with the organization of biological information have grown explosively in the recent years. Bioinformatics is a newly emerging interdisciplinary research area and can be defined as the interface between biological and computational sciences. Thus, the people working in this field in most cases either have a training in biology or computer science, and they learned about the other field by dealing with problems or using the tools of the other one.

So improving an existing old system is important to all biologists, because they need to perform all their functions in more efficient way and using a friendly user interface.

## 1.3 System Objectives

This system strives to accomplish the following:

1. Improving the interface of the old twin gene system.
2. Improving the system performance and speed.
3. Old system testing and error checking and correction for errors exists in the old system.
4. Changing the way of displaying the results of some processes (such as global process)
5. Adding new processes to be performed by the new system such as
  - Importing Blast results from Internet, instead of running blast locally.
  - Interpreting character representation of genes to proteins.
  - Creating the twin gene by assembling the similar genes resulted from grouping.
  - Making global alignment according to score, instead of making it according to similarity percent.

## 1.4 The Current System

The current system serves as computerized biological system, that solves a lot of actions that cannot be done manually, but it focused on computerizing the biological processes more than it focuses on the performance of the system.

In addition, the interface of the system is not comfortable for the user, because it depends on displaying the output on external files, so this was inconvenient for the user, also the time needed for the output to be displayed was very huge, and sometimes the system became idle when comparing large number of genes.

The following table represents comparison between the current system and the new twin gene system:

**Table 1-1: Comparison Between the Current and the new system**

	<b>Old System</b>	<b>New Twin Gene System</b>
<b>Time</b>	30 minuet for Multiple Alignment process	5 minuet for the same process
<b>Efforts</b>	Needs more efforts . The user should determine the paths manually and should open files to see the results.	Less efforts. Facilitate the gene importing And the output displaying.
<b>Accuracy</b>	Wrong results for Grouping process.(ccl4 has 1 group)	Right results. (ccl4 has 2 groups)
<b>Interface</b>	Uncomfortable and unfriendly user interface.	Convenient

## 1.5 The Report Outline

This report shows all important points related to the project, this documentation helps the reader to get comprehensive view about the project. We have tried to present a documentation that can be understood easily by dividing the documentation into a set of subject related parts.

The report consists of nine chapters. Each chapter discusses a subject related to the project, and these chapters are presented according to the logical relations of subjects.

Chapter One: “Introduction”; gives overview about the project, project importance, and Twin Gene system objectives.

Chapter Two: “System planning”; presents system risks, development requirements, cost estimation, operational requirements and time scheduling

Chapter Three: “System Requirements”; it includes definition, analyses and specification.

Chapter Four: “Initial Testing ”; includes the testing that will be made on the previous old system in order to conclude the new system requirements

Chapter Five: “System Design”; consists of the system design model which includes the data flow diagram, structural chart and the data and control analysis.

Chapter Six : “System Implementation”; presents general view; prepares the platform of the system.

Chapter Seven :” System Testing”; shows how the system is going to be tested.

Chapter Eight: “Future Work and Conclusions”; which is in the last chapter, it shows the suggestions for the future development, and points out the conclusions.



Chapter Nine: “Deployment and Maintenance ”; This chapter describes the establishment of the environment that the system will work in, process of deployment and the maintenance plan.

# Chapter Two

## System Planning

### 2.1 Overview

This chapter discusses the system planning concepts such as system risks, development requirement, operational requirements and time scheduling.

### 2.2 System Risks

System risks could be as follow:-

1. Since this system comes out of the reengineering process for a pre-constructed system software , some requirements and models may be restrictive .
2. Since the system processes large sequences of characters(Genes sequences) along with huge Databases, performance metrics (time and size) must be taken into account as a main goal .
3. Since the old system couldn't process a large number of genes, system results may not be clear.
4. The development process may be interrupted by electricity failure.

### 2.3 Development Requirements

Development Requirements includes hardware resources, software resources and the human resources.

#### 2.3.1 Hardware Development Resources

A computer is needed to implement the system; this computer should have the following preferred specification :

Requirement	Required	Recommended
Processor	PC with a Pentium II processor, 450 MHz	Pentium IV, 2.6GHz
Ram	64 MB	256 MB for XP Professional and Visual studio

Hard disk	4 Giga Byte	40 Giga Byte
Printer	Canon BJC-2000	Laser Jet
Video card	800 x 600, 256 colors	High Color 16-bit
Modem	56 Kbps Dial-up	56 Kbps Dial-up

Table 2.1 Hardware development resources

### 2.3.2 Software Development Resources

The PC computer should contain the following operating system and software:

Operating system	Windows XP professional or Windows 2000
Programs	Visual Studio.NET Microsoft Office XP Painter Internet Explorer Blast

Table 2.2 Software development resources

### 2.3.3 Human Development Resources

- Project Team consists of 3 students, they are:-
  - Manar Hammoury
  - Ruba Hirbawe
  - Ruba Sultan
- Project Supervisor:
  - Dr. Mahmoud Saheb
- Co Supervisor:
  - Dr. Yaqoub Ashhab

## 2.4 Cost Estimation

This part contains the estimation of the development costs and the implementation costs.

### 2.4.1 Development Cost

#### Hardware Costs:

Hardware Component	Cost
Pentium IV, 2.6 GHz	750\$

Table2.3 Development Hardware Costs

#### Software Costs:

Software Component	Cost
Windows XP	299 \$
Visual Studio.NET	630\$
Total	929\$

Table 2.4 Development Software Costs

#### Human Resources Costs:

Number of Developers	Work Hours/day (100 days)	Cost per hour for one Developer	Total
3	3	20\$	18000\$

Table 2.5 Development Human Costs

$$\begin{aligned}\text{Total Cost for the Project} &= 750\$ + 929\$ + 18000\$ + 500\$(\text{other expenses}) \\ &= 20179\$\end{aligned}$$

## 2.5 Operational Requirements:

**Required Processor :** Pentium 350 MHz

**Recommended Processor :** Pentium 350 MHz or faster

**Required RAM :** 160 MB

**Recommended RAM :** 160 MB or higher

**Operating System :** Windows XP

**Programs :** .Net framework 1.1 , Blast and Microsoft Internet Explorer

## 2.6 Time Scheduling

Developers started studying the previous system , testing it to find out the reengineering needs and analyzing the new requirements .

The system development time is about 15 weeks. These weeks are organized and arranged in a manner not to exceed the dead line of each task. The following figure depicts the managed time weekly:

Weeks Tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Initial Planning	█	█													
Requirements specification, & analysis			█	█	█										
System Planning				█	█	█	█								
Initial Testing					█	█	█	█	█	█					
System Design							█	█	█	█	█	█	█		
Software implementation								█	█	█	█	█	█	█	
Testing											█	█	█	█	█
Documentation			█	█	█	█	█	█	█	█	█	█	█	█	█

Figure2.1 Time Scheduling

# Chapter Three

## System Requirements

### 3.1 Overview

This chapter describes the requirements of the modified Twin Gene system; it presents requirement definition, and specification.

### 3.2 Main Requirements:

The system requirements in this system are categorized into three main parts, these requirements are:

1. System testing and error checking.
2. System validation .
3. System enhancement.
  - System performance improvement
  - System interface improvement
4. Adding new capabilities to the system.

#### **Main requirements specification:**

##### ▪ **System Testing and error checking:**

The old system has to be tested using black box testing, in order to find out whether the system works correctly or not.

Because our objective is to improve the old system, this testing seems very important point, since no change can be made without ensuring that the system is working correctly.

In addition, when this testing finds out errors in some process, its our functionality to validate them.

- **System validation:**

When system testing provide errors in some place in the old program, the specific location of the errors will be determined, then analysis will be made in order to create a solution that corrects these errors.

Validation that will be made depends on the kind of errors obtained from system testing and error checking.

A modification will be made to some processes that contained errors in the old system.

- **System enhancement**

The major functionality that must be done within this project, is to enhance the old system in more than one field.

The enhancement depends on the performance of the old system, by improving the efficiency in general, and increasing the speed of performing some processes.

In addition there will be an enhancement on the interface of the system.

- **System performance improvement**

The previous twin gene system has to be improved in order to allow the user to perform all the processes in efficient manner, so the system will be changed to produce very high quality output with smaller effort and less time.

- **System interface improvement**

The system is required to be user friendly interface, that helps the user to perform his tasks easily.

The system will have menus, buttons, and other visual component that help the user to complete the processes easily.

The system also will provide very comfortable display for the results of the processes.

- **Adding new capabilities to the system.**

The system is required to allow the user to perform new functionalities that make the system more wide ranging, and can support all the biologist needs.

These function will be explained in detailed in the functional requirement section. Also the system will provide a new nonfunctional requirement that make the system extensively used. These requirements will be explained in the following sections.

- Functional Requirements
- Non-functional Requirements

### **3.3 Functional Requirements:**

We will illustrate the functional requirements through this section. Functional requirements definition and functional requirements specification will be considered.

#### **3.3.1 Functional Requirements Definition**

The functional requirements involve accomplishing the following requirements:

1. Determining the location of the query gene and the database location using file browsing and tree view ,instead of writing its path .
2. Allowing the user to write query on a text field or to import it from any storage media.
3. Displaying the aligned query sections with sections of selected EST produced from blast results in details.
4. Displaying the output of multiple alignment more efficiently .
5. Displaying the output of single alignment more efficiently .
6. Allowing the user to select definite number of headers easily.
7. Allowing the user to deselect the unneeded headers.
8. Enabling the user to visit the NCBI website to view different databases of genes .
9. Enabling the user to import an external BLAST output file to perform Multiple Alignment on it .
10. Displaying the global alignments output using colored bars.
11. Classifying the aligned EST's into suitable groups .
12. Composing the gene sequence which is expected to be the Twin Gene .
13. Translating each resulted query to its analogous protein.
14. Displaying the grouping results more clearly .



### 3.3.2 Functional Requirements Specification

#### 3.3.2.1 Determining the location of the query gene and the database location using file browsing and tree view

**Software project Twin Gene /3.3.2.1**

**Function:** Allow a user to determine the location of query and data base through browsing .

**Description:** In this part of the system, the user can determine the path of the query gene using file browser or using the tree in front of him, instead of writing it manually.

**Inputs:** Specifying the file path .

**Source:** . The place that contains the file, it may be on some location on drive C

**Outputs:** A textbox that will be filled by the query file path.

**Destination:** A textbox that will be filled by the query.

**Requires :** none

**Pre-Condition:** Test the existence of the file.

**Post-Condition:** None

**Side-effects:** None.

**Definition: Software Project / Twin Gene /3.3.2.1**

### **3.3.2.2 Allow the user to write query on a text field or to import it from any storage media.**

#### **Software Project Twin Gene / 3.3.2.2**

**Function:** Allow a user to write query on a text field or to import it from any storage media.

**Description:** The user can get the query gene from any place and copy it on a text field, also in this part, the query gene can be browsed and imported from local storage media.

**Inputs:** The query gene format.

**Source:** Either the internet or local storage media contain the query gene.

**Outputs:** A text field will be filled with the query gene format .

**Destination:** This text representation of the query gene will be used for the multiple alignment and other processing .

**Requires :** Getting the query gene format.

**Pre-Condition:** special text representation for the query gene .

**Post-Condition:** testing the incoming query.

**Side-effects:** none.

**Definition: Software Project / Twin Gene/3 .3.2.2**

### 3.3.2.3 Displaying the aligned query sections with sections of selected EST produced from blast results in details

#### **Software Project / Twin Gene / 3.3.2.3**

**Function:** Displaying the aligned query sections with sections of selected EST produced from blast results in details.

**Description:** The output of blast will be displayed to user in details, so the query gene will be displayed with sections of similar EST's, in order to allow the user to determine the similar sections visually .

**Inputs:** Name of the file that contains the query and name of the file that contains the output of the blast program. The names should be specified with full path.

**Source:** The place that contains the two files, it may be on some location on drive c.

**Outputs:** The list of each query with its similar EST.

**Destination:** A list Displayed on the multiple alignment form.

**Requires:** Correct file path.

**Pre-condition:** Test the existence of the file.

**Post –condition:** None.

**Side effects:** None.

**Definition: Software Project / Twin Gene/3 .3.2.3**

### 3.3.2.4 Displaying the output of multiple alignment in more efficient way.

**Software Project / Twin Gene / 3.3.2.4**

**Function:** Displaying the output of multiple alignment in more efficient way.

**Description:** In this part of the system, the output of the multiple alignment process will be displayed more quickly than before .

**Inputs:** multiple alignment results..

**Source:** The array stored temporarily after making multiple alignment..

**Outputs:** The query gene and its similarities .

**Destination:** A rich text box will be filled with the output .

**Requires:** correct result from multiple alignment .

**Pre-condition:** none .

**Post –condition:** None.

**Side effects:** None.

**Definition: Software Project / Twin Gene/3 .3.2.4**

### 3.3.2.5 Displaying the output of single alignment in more efficient way.

#### **Software Project / Twin Gene / 3.3.2.5**

**Function:** Displaying the output of single alignment in more efficient way.

**Description:** In this part of the system, the output of the single alignment process will be displayed more quickly than before .

**Inputs:** Single alignment results..

**Source:** The array stored temporarily after making single alignment..

**Outputs:** The query gene and its similar gene .

**Destination:** A rich text box will be filled with the output .

**Requires:** correct result from single alignment .

**Pre-condition:** Test the existence of single alignment output .

**Post -condition:** None.

**Side effects:** None.

**Definition: Software Project / Twin Gene/3.3.2.5**

### 3.3.2.6 Allow selection of definite number of headers easily

**Software Project / Twin Gene / 3.3.2.6**

**Function:** Allow selection of definite number of headers easily.

**Description:** In this part of the system, the user can select the EST's to make multiple alignment with them, so the user can select large number without checking each one manually.

**Inputs:** Number of needed genes.

**Source:** User Determined.

**Outputs:** The selected genes.

**Destination:** The list of headers.

**Requires:** : correct input number from the user.

**Pre-condition:** Test for a correct number.

**Post-condition:** None.

**Side effects:** None.

**Definition: Software Project / Twin Gene/3.3.2.6**

### 3.3.2.7 Allowing the user to deselect the unneeded header.

**Software Project / Twin Gene / 3.3.2.7**

**Function:** Allowing the user to deselect the unneeded headers.

**Description:** In this part of the system, the user can deselect the header of the not required gene and selected before .

**Inputs:** list of headers.

**Source:** previous process .

**Outputs:** The selected genes in the list of headers will be unselected .

**Destination:** The list of unselected headers .

**Requires:** determining to deselect .

**Pre-condition:** None.

**Post -condition:** None.

**Side effects:** None.

**Definition: Software Project / Twin Gene/3.3.2.7**

### 3.3.2.8 Enabling the user to visit the NCBI website to view different databases of genes.

**Software Project / Twin Gene / 3.3.2.8**

**Function:** Enabling the user to visit the NCBI website to view different databases of genes.

**Description:** The user can visit the NCBI web site in order to make use of the updated gene database, or to get a pre-executed BLAST results.

**Inputs:** The name of needed query gene ,database or needed blast results .

**Source:** User determined

**Outputs:** The query representation, database or the blast output .

**Destination:** On web page .

**Requires:** None

**Pre-condition:** On line connection.

**Post -condition:** None

**Side effects:** None.

**Definition: Software Project / Twin Gene/3.3.2.8**



### 3.3.2.9 Enabling the user to import an external BLAST output file to perform Multiple Alignment on it

#### **Software Project / Twin Gene / 3.3.2.9**

**Function:** Enabling the user to import an external BLAST output file to perform Multiple Alignment on it

**Description:** The user can get a pre-executed BLAST results from internet, and save it on the local PC then use it to make the needed processes.

**Inputs:** determining the location of the BLAST result .

**Source:** Local storage

**Outputs:** The BLAST result .

**Destination:** On locally stored file .

**Requires:** None

**Pre-condition:** None

**Post -condition:** None

**Side effects:** None.

**Definition: Software Project / Twin Gene/3 .3.2.9**

### 3.3.2.10 Displaying the global alignments output using bars.

**Software Project / Twin Gene / 3.3.2.10**

**Function:** Displaying the global alignments output using bars.

**Description:** In this part of the system, the result of the global alignment will be displayed using colored bars to enable the user to determine the groups of genes that are agrees on difference with the query.

**Inputs:** The Name of the file that contains query, name of the file that contains blast output.

**Source:** The place that contains the two files, it may be on some location on drive c..

**Outputs:** The rich text box that contains the query and its similarities from the EST's in the shape of colored bars .

**Destination:** The rich text box of colored bars indicating the query and its similarities.

**Requires:** Existence of files .

**Pre-condition:** Testing the existence of needed file.

**Post -condition:** None.

**Side effects:** None.

**Definition: Software Project / Twin Gene/3.3.2.10**

### 3.3.2.11 Classifying the aligned EST's into suitable groups.

**Software Project / Twin Gene / 3.3.2.11**

**Function:** Classifying the aligned EST's into suitable groups.

**Description:** In this part of the system, the grouping process will be made in different way that it made before, and it divides the genes in the test.out into groups based on correlation of differences with the query gene. This process is accomplished using sophisticated comparisons. It compares each character in each EST with each character in all other EST's.

**Inputs:** The aligned EST's from the multiple alignment stage.

**Source:** The array that contains the EST's.

**Outputs:** The group of each EST in the rich text box.

**Destination:** The rich text box of colored bars indicating the query and its similarities.

**Requires:** Existence of array that contains the EST's .

**Pre-condition:** Testing the existence of array .

**Post -condition:** None.

**Side effects:** None.

**Definition: Software Project / Twin Gene/3.3.2.11**

### 3.3.2.12 Composing the gene sequence which is expected to be the Twin Gene.

**Software Project / Twin Gene / 3.3.2.12**

**Function:** Composing the gene sequence which is expected to be the Twin Gene.

**Description:** After the EST's are classified into groups, a new gene will be constructed to represent each group, this new gene is likely to be the Twin Gene.

**Inputs:** All the classified EST's .

**Source:** The grouping results

**Outputs:** The new composed Twin Gene .

**Destination:** Text field that contains the each group with it EST's and the new composed Twin Gene.

**Requires:** The grouping process

**Pre-condition:** None .

**Post -condition:** None.

**Side effects:** None.

**Definition: Software Project / Twin Gene/3.3.2.12**

### 3.3.2.13 Translating each resulted query to its analogous protein

**Software Project / Twin Gene / 3.3.2.13**

**Function:** *Translating each resulted query to its corresponding protein.*

**Description:** Each new Constructed gene will be translated into its equivalent protein representation.

**Inputs:** The constructed genes .

**Source:** The composing process result.

**Outputs:** The protein representation of the composed gene.

**Destination:** A text field that will contain the new composed genes with their protein representation

**Requires:** Genetic code table

**Pre-condition:** None

**Post –condition:** None

**Side effects:** None.

**Definition:** *Software Project / Twin Gene/3 .3.2.13*

### 3.3.2.14 Displaying the grouping results more clearly

**Software Project / Twin Gene / 3.3.2.14**

**Function:** *Displaying the grouping results more clearly.*

**Description:** The results for grouping process will be displayed clearly, the classified EST's, the new composed gene and the protein representation will be collected to be represented in one obvious form.

**Inputs:** The classified EST's, new composed genes and the protein representation for each gene .

**Source:** The grouping process, composing process and the translation process .

**Outputs:** The protein representation of the composed gene.

**Destination:** An obvious form contain the collection of the grouping results.

**Requires:** None

**Pre-condition:** Correct result for grouping, composing and translation.

**Post -condition:** None

**Side effects:** None.

**Definition:** *Software Project / Twin Gene/3.3.2.14*

### **3.4 Non-Functional Requirements**

The system has to organize non-functional requirements. This organization guarantees powerful and fault tolerant system. a non ambiguous graphical user interface will lead to suitable interaction between the system and its users.

Non-Functional requirements aspects of the system are:

- Friendly user interface
- Fast and easy interaction
- High speed program
- High performance system
- System Dependency
- System Portability

#### **1. Friendly user interface**

Visual Studio .NET is a power full tool for developing system. It provides all the required components to develop a friendly user interface. Menus, Drop down lists, Data grids, Buttons, Labels, Hyperlinks and other graphical components were incorporated among the system to grant intelligible interface.

#### **2. Fast and Easy interaction**

The windows application parts of the system steers users in obvious paths of interaction. Most user operations depend on selecting an option rather than writing it, this provide an easy and fast interaction.

#### **3. High speed program**

The system enables the user to perform all the processes in a short period of time, and take the results of the processes very quickly.

#### **4. High performance system**

The system will provide very high quality output, with less cost.

#### **5. System Dependency**

The system can accept very large number of genes to align them with the query gene, also can accept very long query gene.

#### **6. System Portability**

The system can be run at any platform, and it can be installed on any directory or drive without any problem.



# Chapter Four

## Initial Testing

### 4.1 Overview

In order to modify the previous twin gene system, it must be examined by testing the program units and the user interface .This chapter contains the testing of user interface and the system modules (Multiple Alignment, Single Alignment, Global Alignment and Grouping) .

### 4.2 User Interface Testing

In the interface testing stage from the user view , no errors can be detected but dealing with the system may be uncomfortable as a result of unfriendly user interface. In this testing for the previous twin gene program we test the user interface and we detect some gaps that must be modified.

#### 1. Main page

- To get the database and the query gene files the path must be written by the user.
- The gene must be imported just from a file ,so the user can't give a part of gene from an external file and paste it to enter to some processes.
- Processes sequence is not clear for the user.

#### 2. MSA page

- Selecting of the headers is uncomfortable since it must be manually .
- The parameters of different processes like multiple sequence alignment and grouping exist in the same page.
- To view the output of a process a file must be browsed.
- Since output files are html documents ,it takes a lot of time to view results.
- The Global process doesn't give an obvious general alignment of EST's with the gene.
- The Grouping results don't describe the groups clearly.

- The Grouping doesn't give the expected genes to choose one of them to be the twin gene of the query gene.

### 4.3 Processes testing

In this stage, the black box testing is used to examine the functionality of system processes

- **Multiple Sequence Alignment**

Input	Expected output	Actual output	Notes
Query gene (livin), Select 5 headers, Similarity limit =99%.	The aligned 5 est's of similarity limit more than 99% .	The aligned 5 est's of similarity limit more than 99%.	Output display on html document takes a lot of time
Query gene (livin), Select 100 headers, Similarity limit =90%.	The aligned 100 est's of similarity limit more than 90%.	Incomplete output	Large amount of data makes the output file not responding.
Query gene (pten), Select 20 headers.	The aligned 20 headers with the gene.	None	The system breaks down since it is a long query gene.
Query gene (ccl3), Select 10 headers.	The aligned 10 headers with the gene.	None	The system breaks down for an error

- **Global Alignment**

Input	Expected output	Actual output	Notes
Query gene(ccl4), Select 10 headers	General view of the aligned EST's with the query based on the score value	Alignment of EST's with query based on similarity positions	Same as MSA (The character base was replaced by a block of its color)

- **Single Alignment**

<b>Input</b>	<b>Expected output</b>	<b>Actual output</b>	<b>Notes</b>
Query gene (livin), select one header	Query sequence aligned with the EST sequence	Query sequence aligned with the EST sequence	Output display on html document takes a lot of time
ery gene (ccl4), select one header	Query sequence aligned with the EST sequence	Query sequence aligned with the EST sequence	Output display on html document takes a lot of time

- **Grouping**

<b>Input</b>	<b>Expected output</b>	<b>Actual output</b>	<b>Notes</b>
Query gene (ccl4), Select 40 headers , Minimum percent length=1	2 groups	2 groups	None
Query gene (ccl4), Select 241 headers , Minimum percent length=1	2 groups	3 groups	None
Query gene (ccl4), Select 241 headers , Minimum percent length=3	2 groups	4 groups	None
Query gene (ccl4), Select 100 headers , Minimum percent length=3	2 groups	2 groups	None
Query gene (ccl4), Select 50 headers Minimum percent length=10	2 group	3 groups	None

Query gene (ccl3), Select 40 headers , Minimum percent =1	2 groups	1 groups	None
Query gene (ccl3), Select 250 headers , Minimum percent =3	2 groups	4 groups	None
Query gene (ccl3), Select 10 headers Minimum percent length=7	2 groups	None	The system breaks down.
Query gene (ccl3), Select 100 headers Minimum percent length=3	2 groups	1 group	We change the copy of ccl3,so the program continue the execution
Query gene (ccl3), Select 50 headers Minimum percent length=4	2 groups	4 groups	None
Query gene (ccl3), Select 50 headers Minimum percent length=10	2 groups	2 groups	None
Query gene (livin), Select 50 headers Minimum percent length=4	1 groups	1 group	None
Query gene (livin), Select 60 headers Minimum percent length=6	1 groups	1 groups	None



- Global alignment arrays

- score: array of integers, used to store the score value for each EST

```
Dim score As Integer() = New Integer(q - 1) {}
```

**Array of integers**

<b>1</b>	50
	170
<b>q-1</b>	

- startend: jagged array, used to store the start and end points of related blocks to each EST

```
Dim startend As Integer()() = New Integer(q - 1)() {}
```

<b>1</b>	1	100	130					
	20	80						
	1	600						
<b>q-1</b>								

- color: array of colors, used to store the color of each EST depending on its score

```
Dim c As Color() = New Color(q - 1) {}
```

**Array of colors**

<b>1</b>	Black
	Red
<b>q-1</b>	

- Grouping arrays

➤ dif: two dimensional array of strings, used to store the EST characters that differs from query characters.

```
Dim dif As String(,) = New String(no, Len(stt)) {}
```

		Array of strings						EST Length
1	A	T						
	C							
	No of EST's							

➤ query: array of characters, used to store the composed twin gene resulted from grouping process.

```
Dim query As Char() = New Char(Len(stt)) {}
```

		Array of characters
1	A	
	T	
	T	
	C	
q-1		

## 2. Records

New data record is created, because the program is often required to process much more data, especially in grouping algorithm.

- › Twins record contains the query field, and it is used in translation process

```
Structure twins
  Dim query As String
  Dim protein As String
End Structure
```

## 3. Files

Two files are required to be created, in order to reengineer all program parts to use a common files, to avoid data inconsistency and file problems.

- › Queryfile: A text file that created to contain the character representation of the query gene, and this file is used as an input to blast program, and used in multiple alignment and global process.

```
Dim queryfile As File
Queryfile.createText(Query.Text.ToCharArray)
```

```
>ccl4
CTTCTGAGTTCTGCAGCCTCACCTCTGAGAA
AACCTCTTTTCGACCAATACCATGAAGCTCT
GCGTGACTGTCCTGTCTCTCCTCATGCTAGT
AGCTGCCTTCTGCTCTCCAGCGCTCTC...
```

- › Fixdata: A text file created to store the database for genes.

```
Dim fixdata As File
Fixdata.createText(data)
```

```
>gb|UG4387618|Hs#S4387618
AGENCOURT_6859683 Homo sapiens cDNA, 5' end
/clone=IMAGE:5928696 /clone_end=5' /gb=BQ072175
/gi=19901221 /ug=Hs.82932 /len=1076
CAGTAGCAGCGAGCAGCAGAGTCCGCACGCTC
CGGCGAGGGGCAGAAGAGCGCGAGGGAG
CGCGGGGCAGCAGAAGCGAGAGCCGAGCGC...
```



## 5.3 Data needed for the system

After doing some kinds of surveying and after studying the old twin gene system we deduced the data that are needed to operate the system :

### 5.3.1 System Files:

- **The Database of Genes**

This database contains the header and the sequence of each gene, which is an input for blast program, to make comparison with query gene.[1]

- **The Query Gene**

This is a text representation of a gene, starts with '>' and the name of this gene, followed by a sequence of characters.

This gene can be imported as a file, or can be written on a text field .

This gene is used as input to blast program, to make alignment with each gene sequence in database.

- **The BLAST output file:**

The blast output is a file that results from blast program which can be run through the system, or imported as external file. It contains the headers of the EST's, the query gene aligned with each EST and information about alignment process.

This file is used by multiple alignment, single alignment and global alignment processes. [2]

For more information about BLAST output file, go to the old project.

### 5.3.2 System Parameters

- **The Parameters of blast program**

- The expectation value, which determines the score of the EST's.

This value can be selected from list of values in the system.

- **The Multiple Alignment parameters**

- The similarity limit: The percent of how much the query is similar to each EST.
- The EST section length: The minimum length of the EST with opposite direction to be taken into account in the alignment process.

- **The Grouping Parameters**

- Minimum percent: It is the length of the EST from the length of the query, “what is the minimum number of characters in EST that differs from the query gene”.
- Significant start: The number of characters to be discarded when count the differences.

## 5.4 Design Models

Design models are representative models of the software system that describe the system in the mode of figures and designs which give the reader knowledge of the system at a glance and proceed without involving through the system details or testing. These models include General block diagram, Data flow diagram, Structure Chart, and Flowcharts of subsystems and processes.

### 5.4.1 System Block Diagram

This diagram shows an overview of the subsystems that compose the Twin Gene system and the relationships between each others . This diagram shows only the subsystems names without details about the relationships.

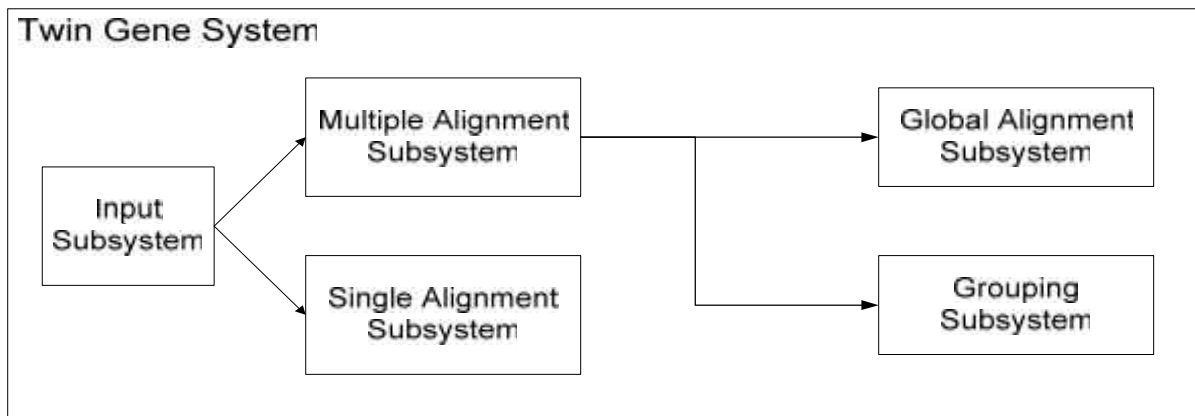


Figure 5.1 System Block Diagram

### 5.4.2 Data Flow Diagram (DFD)

Data flow is concerned with designing a sequence of functional transformation that convert the system inputs into the required outputs. Twin Gene system data flow diagram starts with importing the query sequence from its location; its location may be a file or a text field and so, use Formatdb (to format protein or nucleotide source databases before these Databases can be searched by BLAST) ,then use BLAST program to compare the query gene sequence with Database EST's, after that multiple alignment process will be done; the query gene and similar EST's in BLAST output file are aligned ,ordered and displayed on a color supportive field so that its clear to view similarities and differences through EST's . The flow of the process may continue with aligning the resulted EST's from multiple alignment process globally by drawing colored descriptive bars , or it may continue with dividing the resulted EST's into groups based on correlation of differences with the query gene . Since both of global alignment and grouping processes depend on the multiple alignment results, its clear that it should be performed before each one of them .

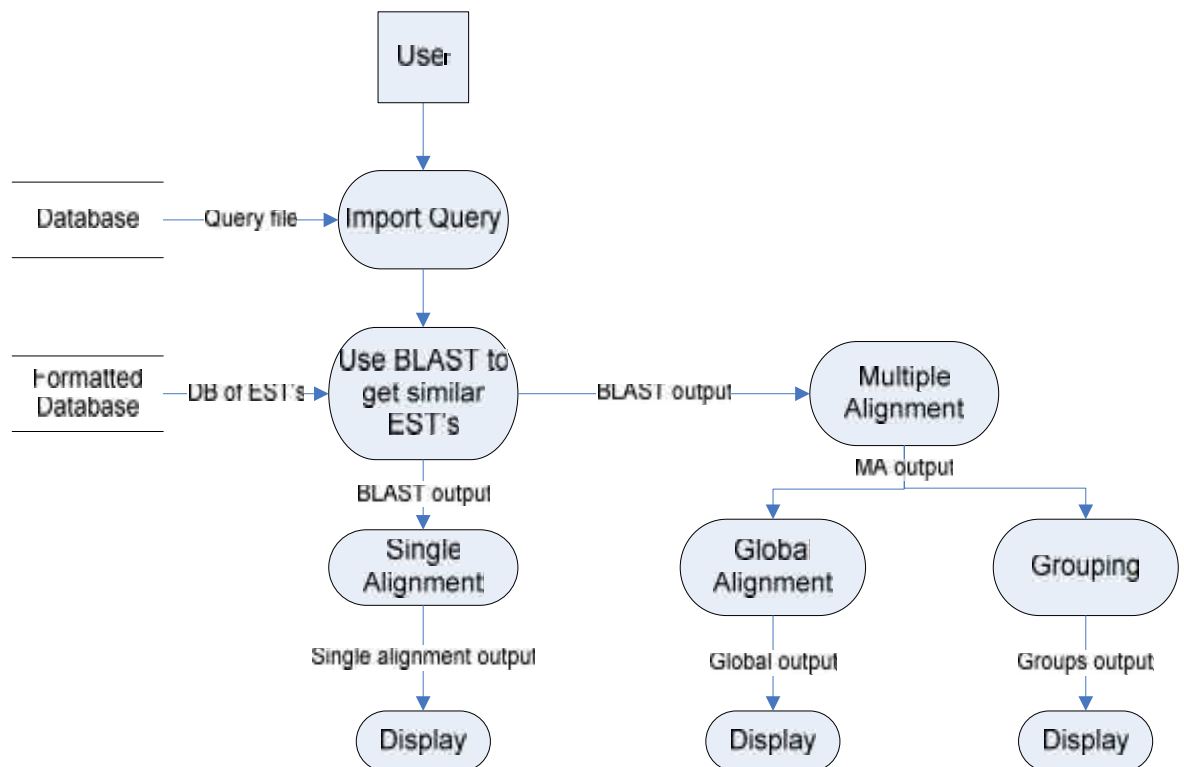


Figure 5.2 Data Flow Diagram

### 5.4.3 System Structure Chart

This diagram shows more specific information about the main components of the system, by showing how a function is realized by a number of other functions .

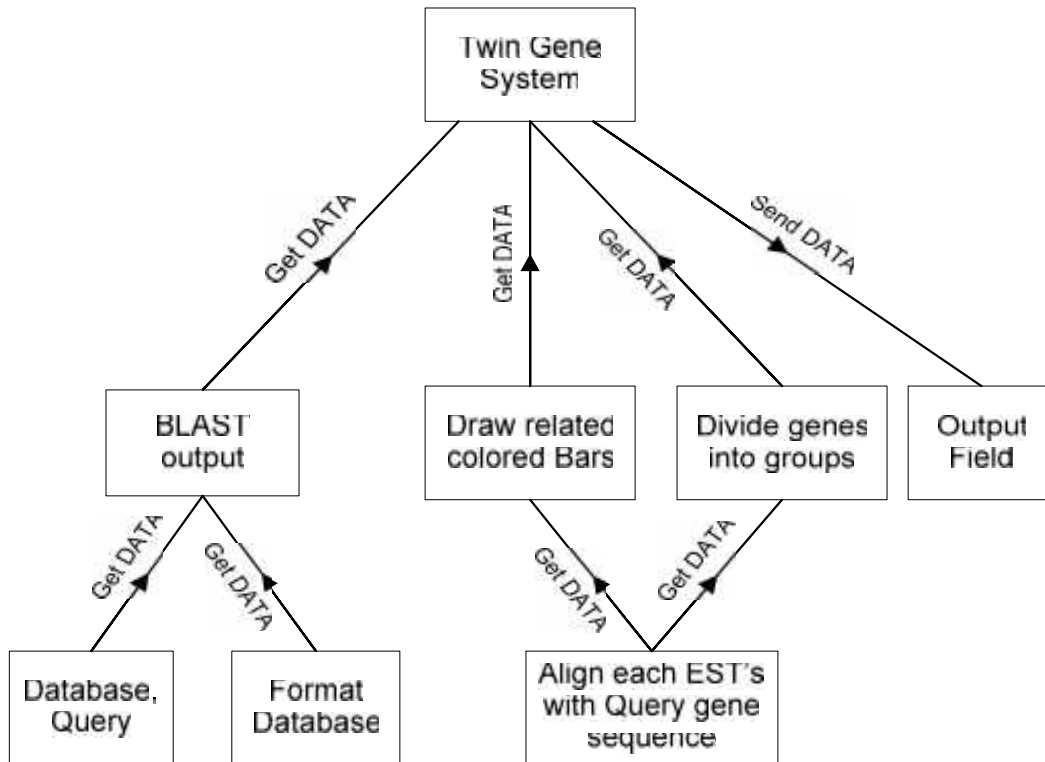
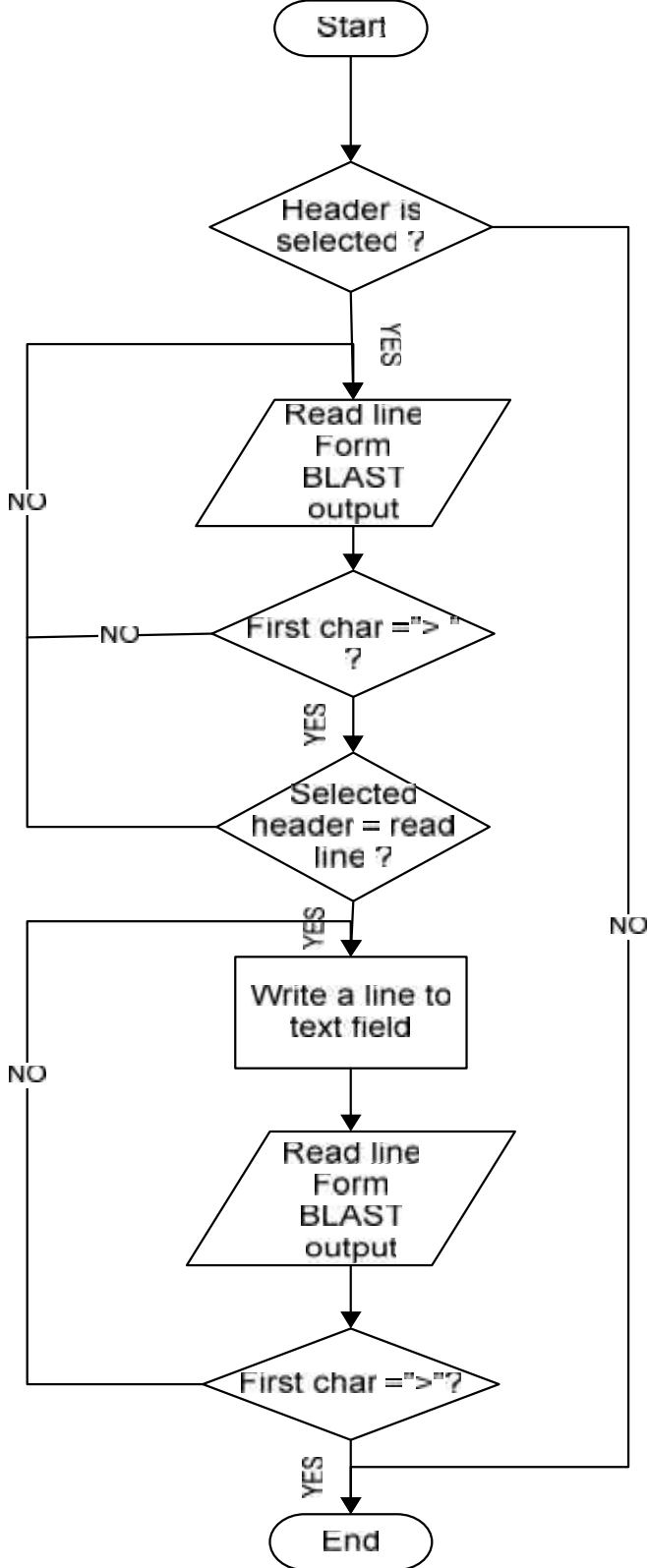


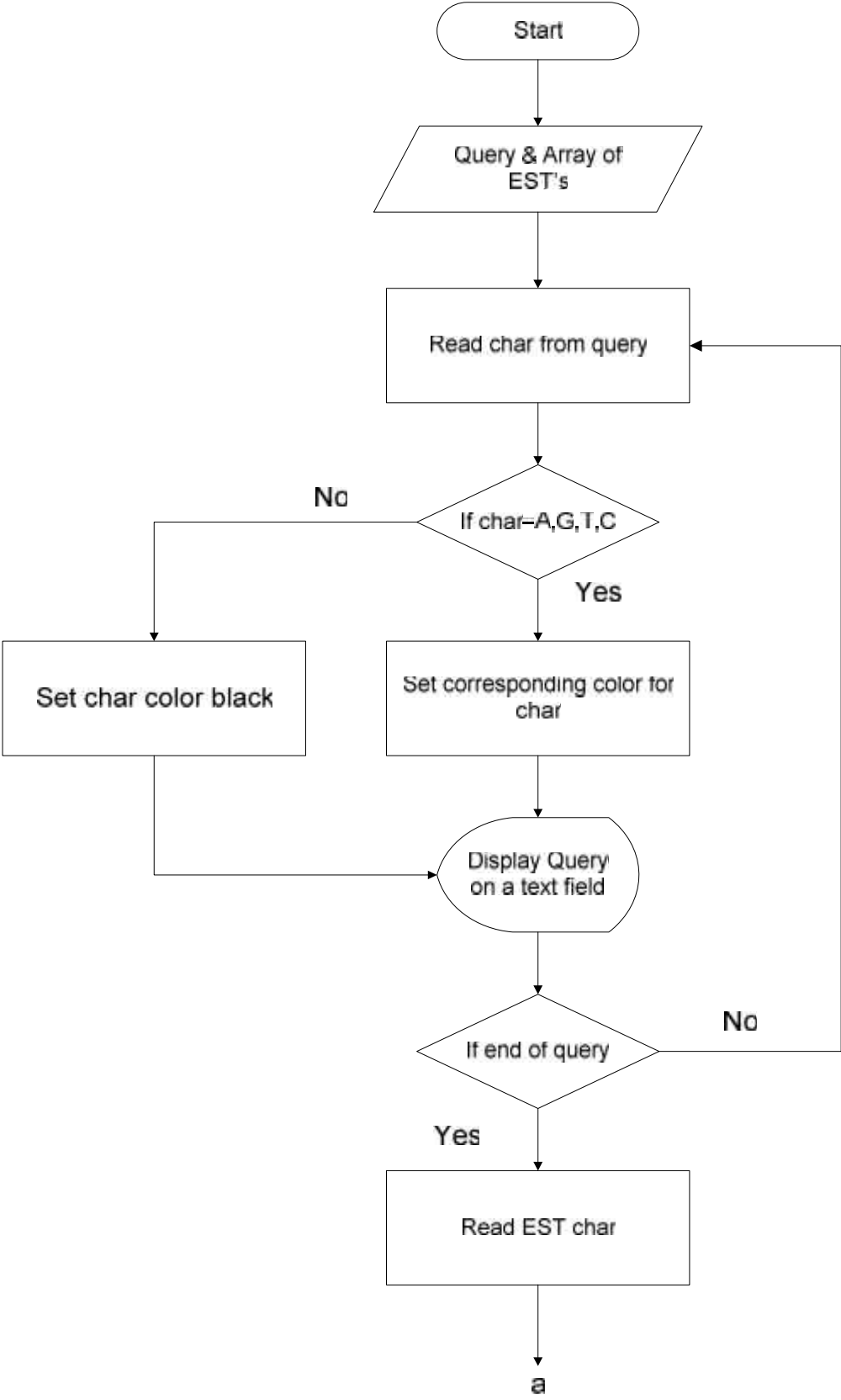
Figure 5.3 System Structure Chart

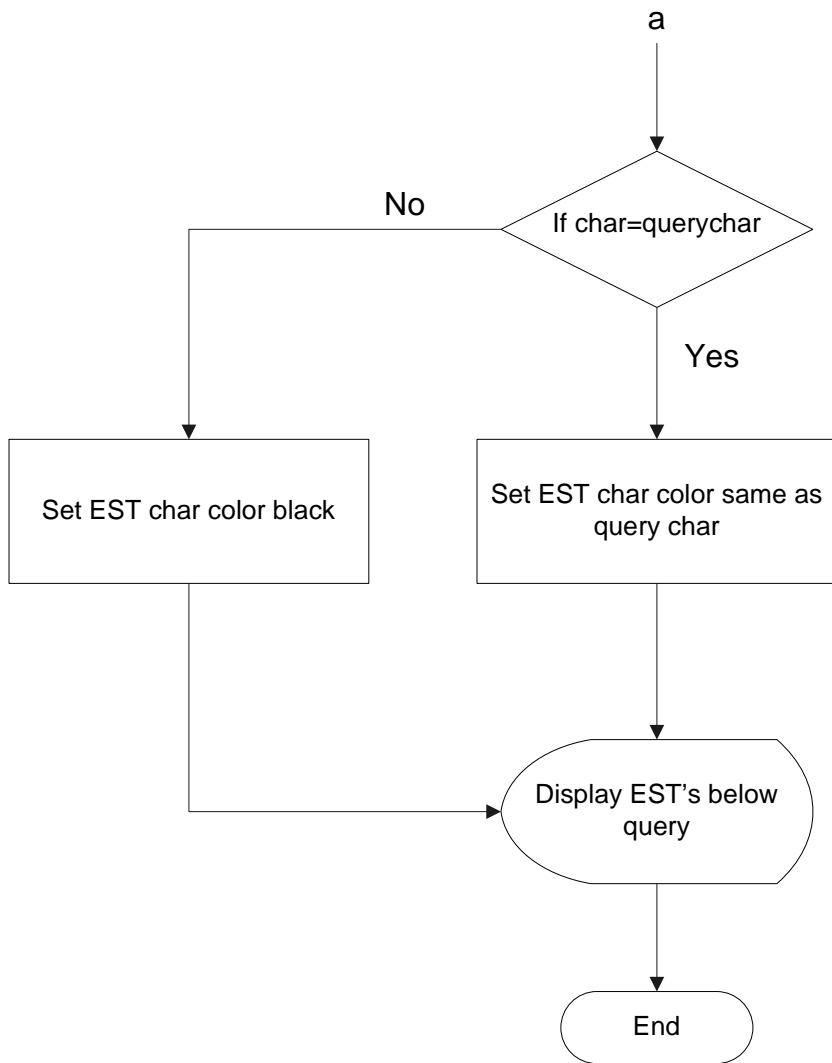
5.4.4 System Flowcharts

5.4.4.1 Display the alignment of query with EST



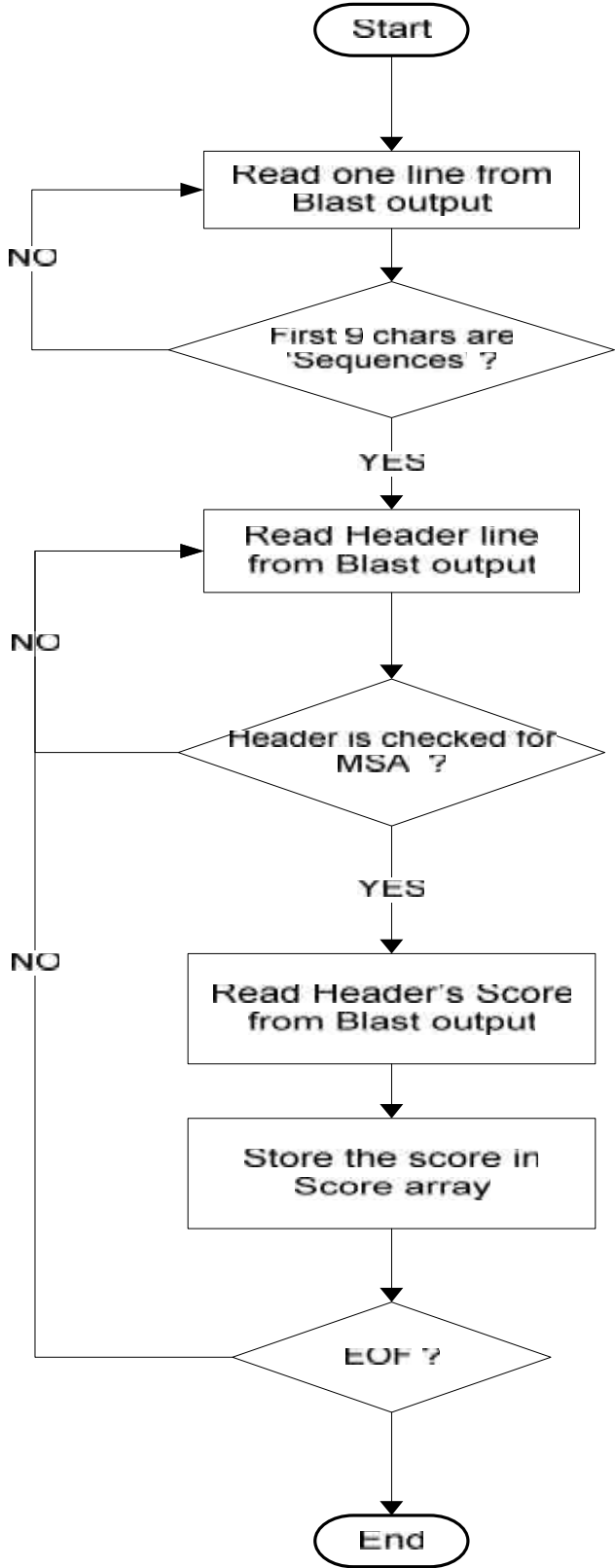
5.4.4.2 Multiple Alignment result





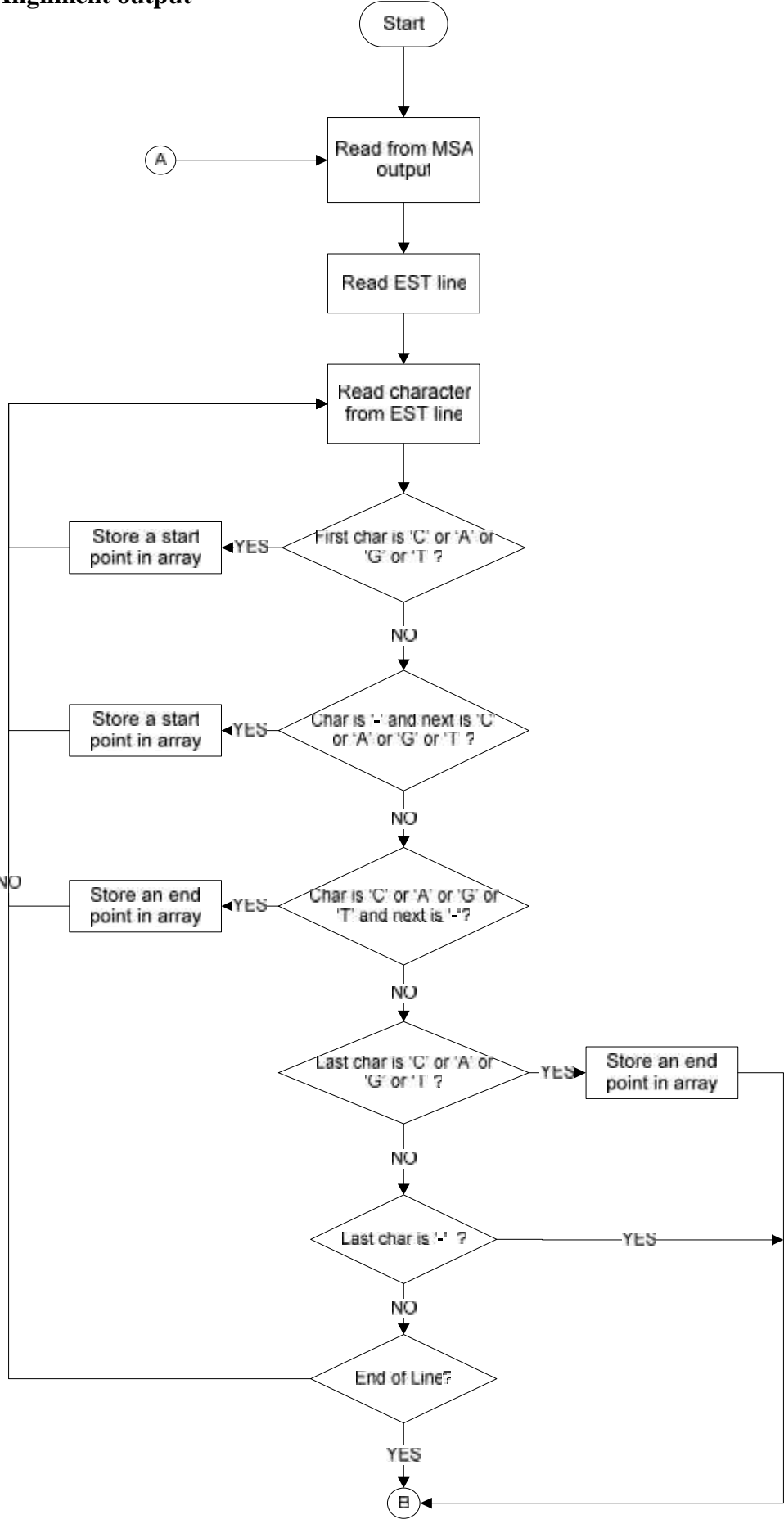
5.4.4.3 Global Alignment

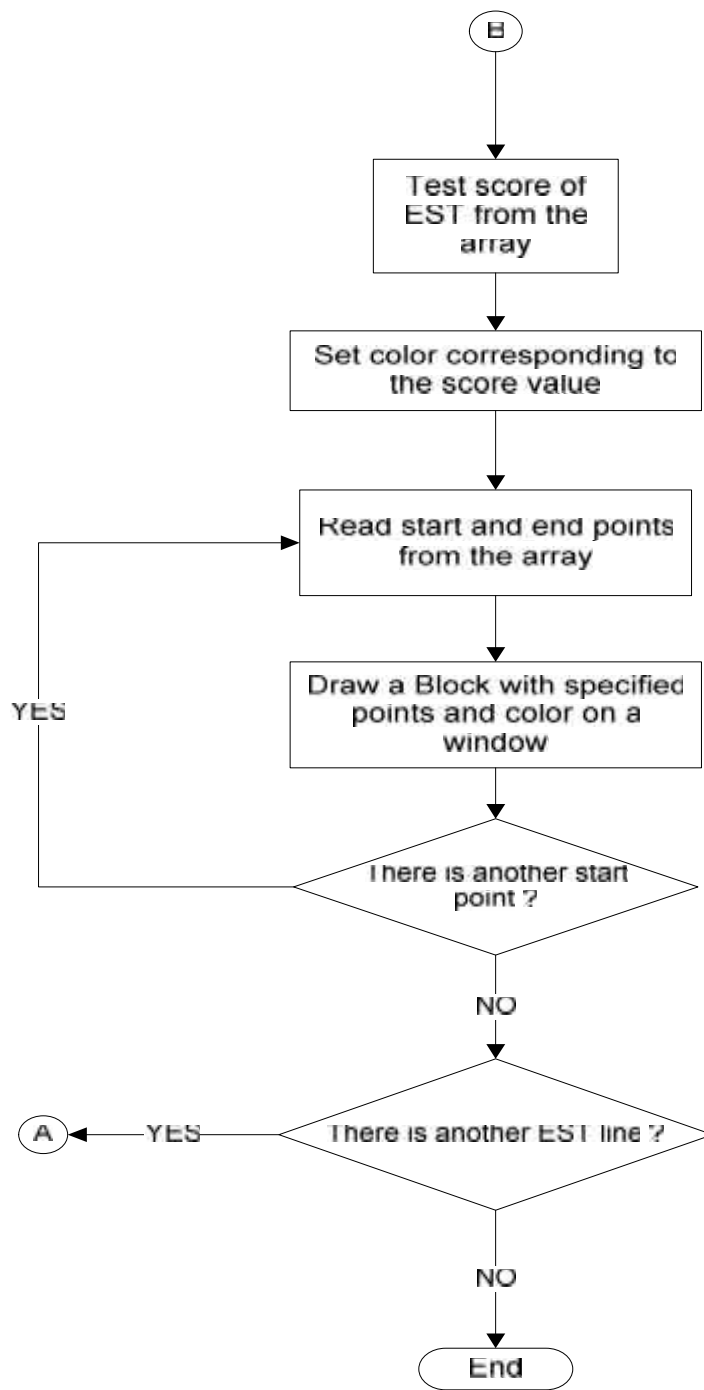
Score reading





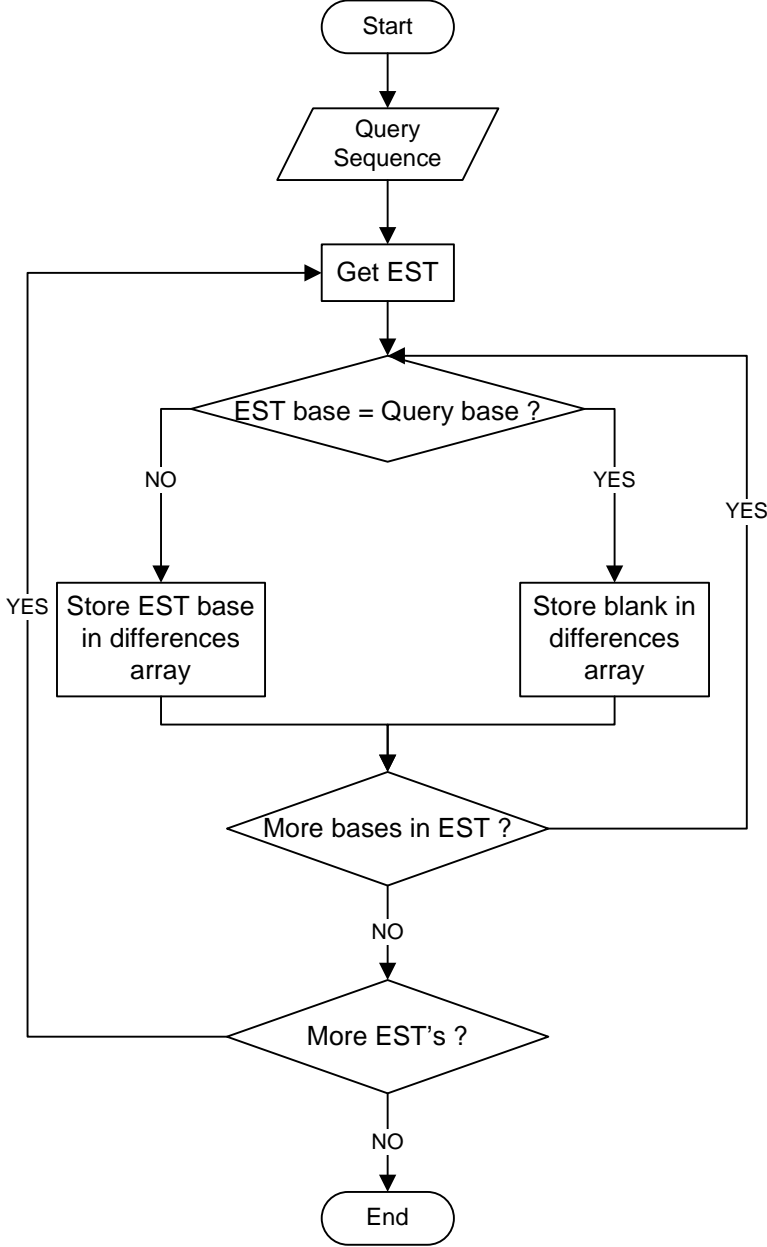
**Global Alignment output**



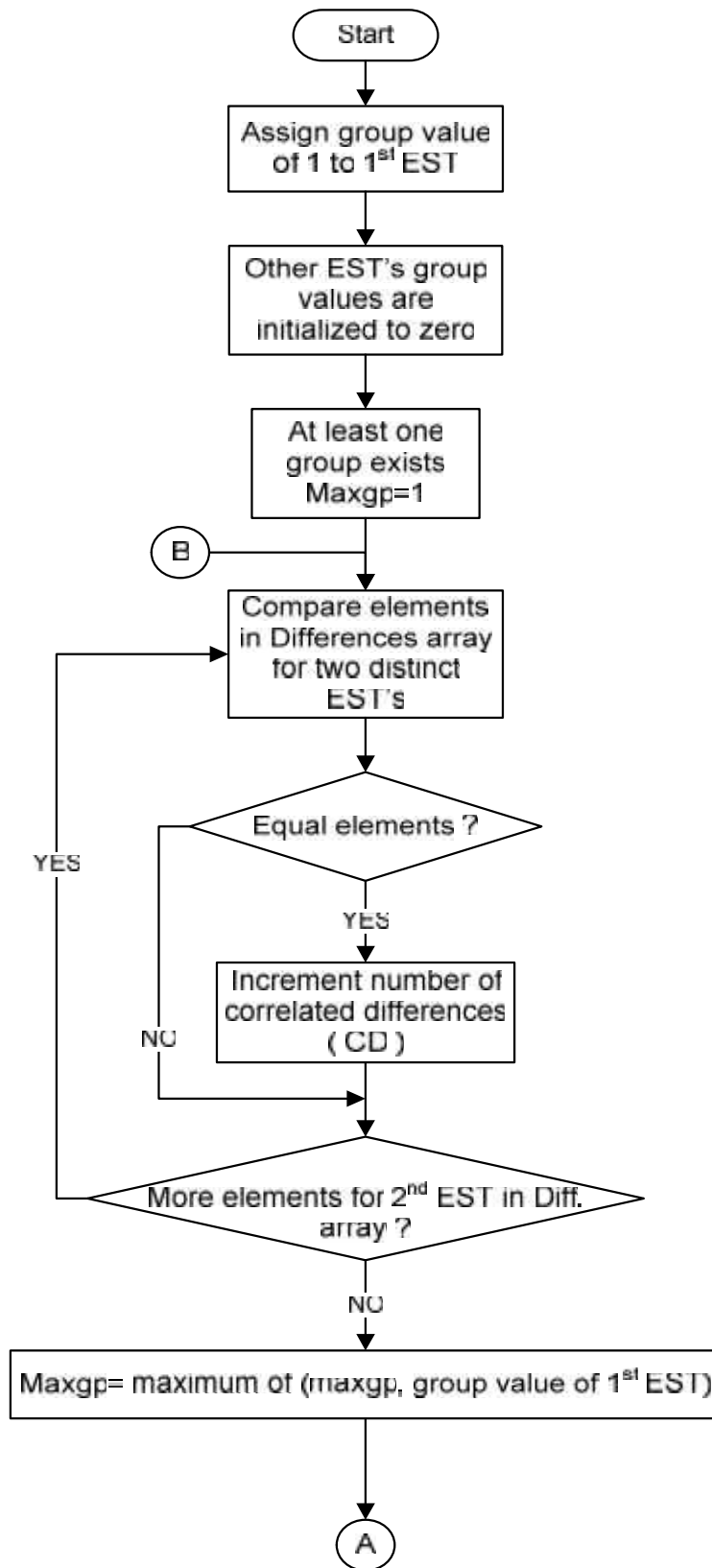


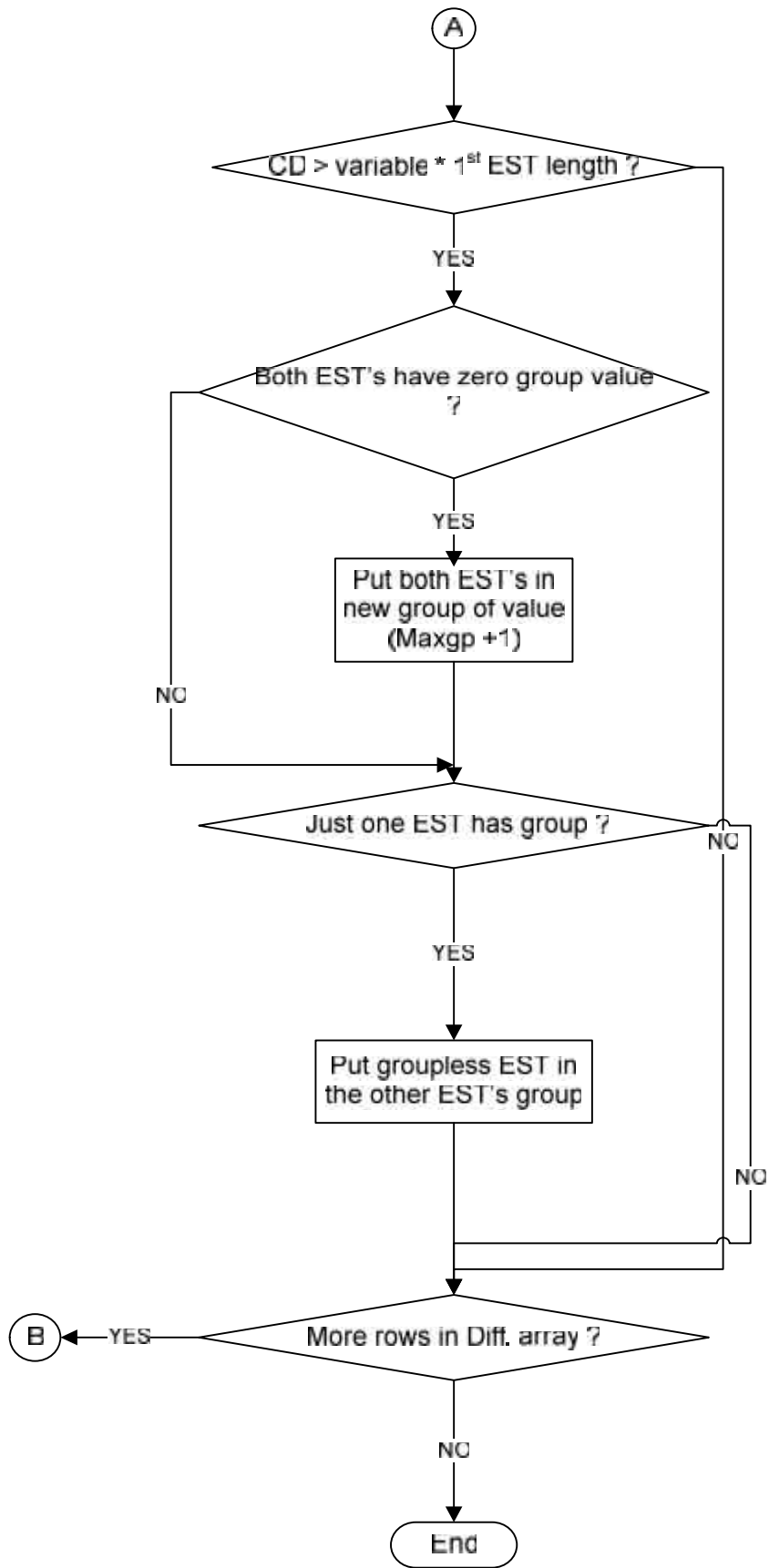
5.4.4.4 Grouping

Differences array construction

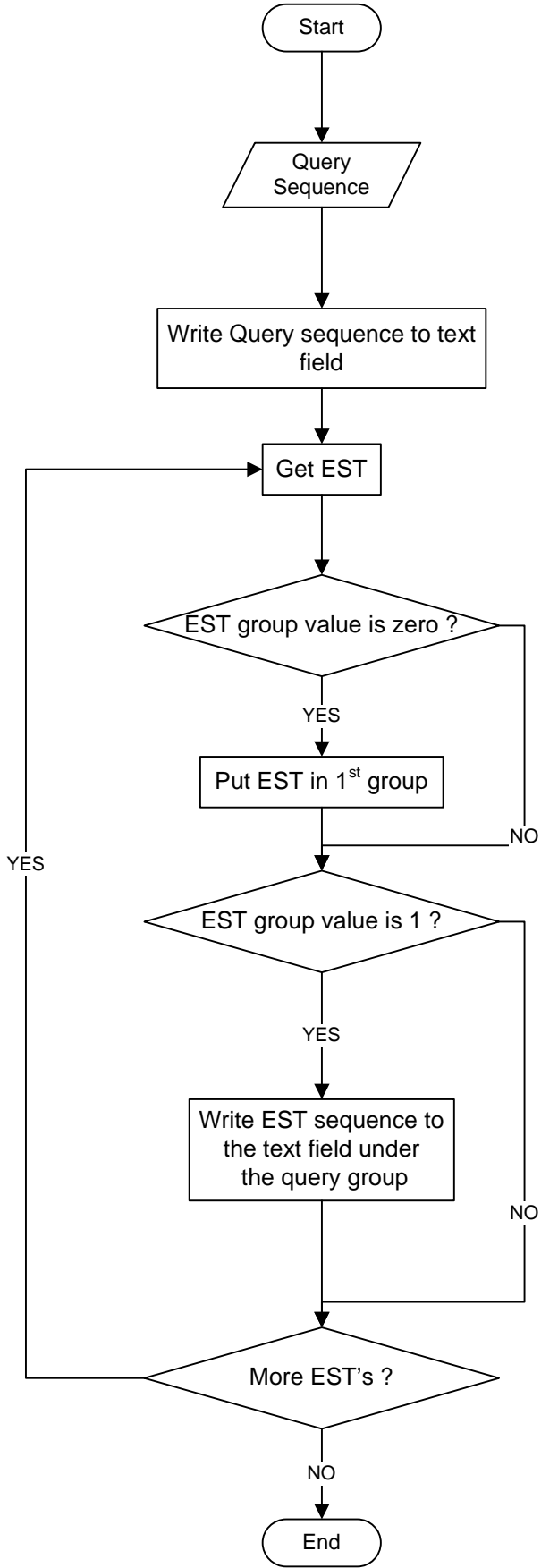


## Classifying EST's into groups

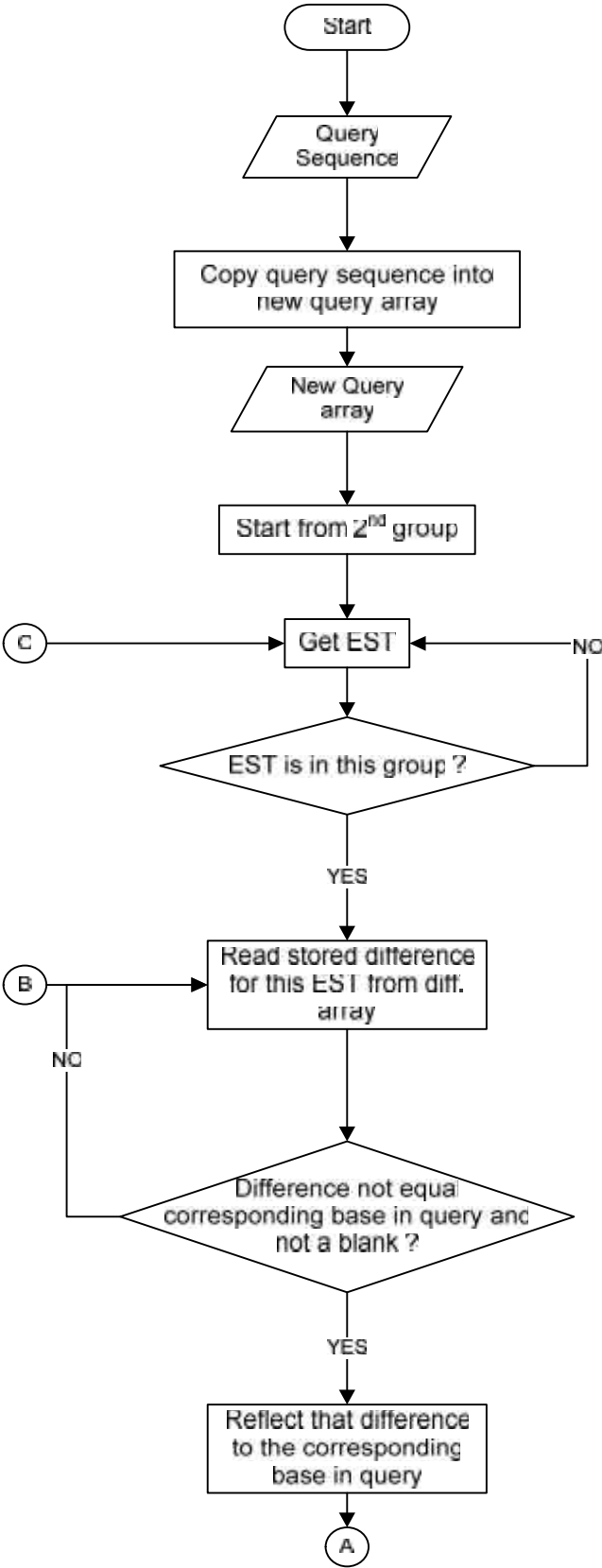


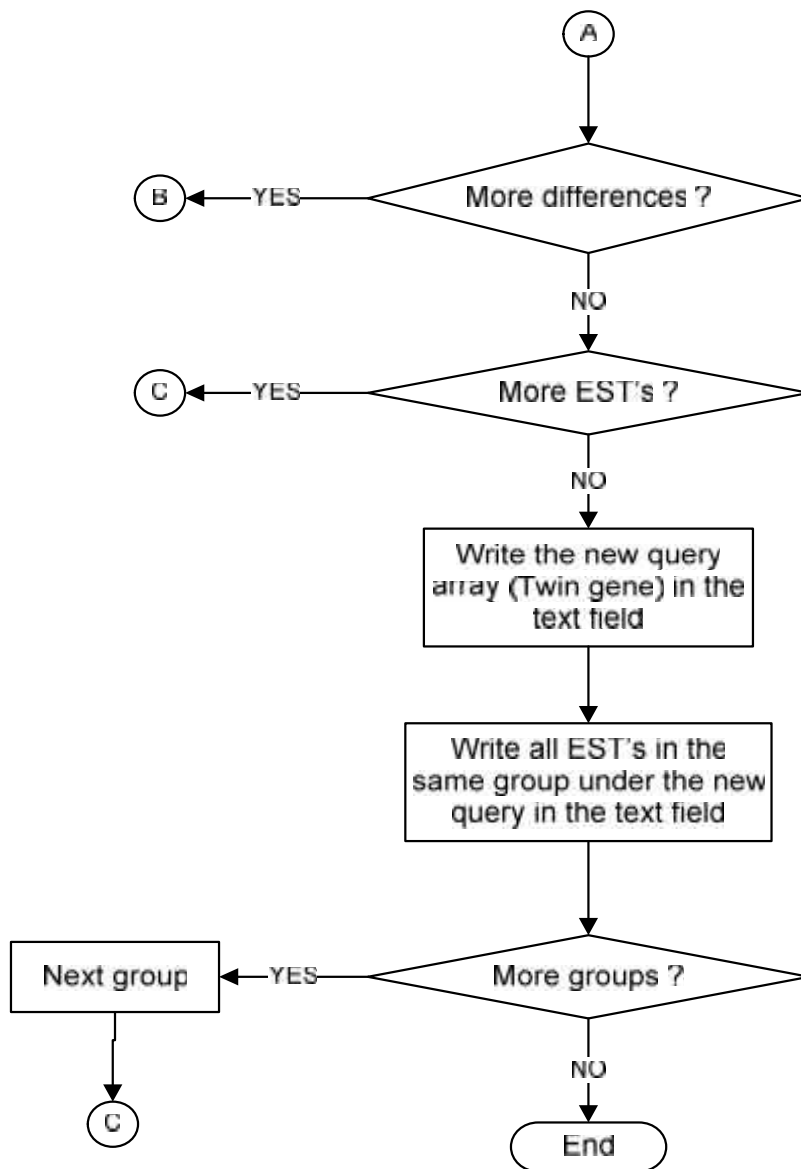


**Writing the query gene and correlated EST's**



**Composing and displaying the new Twin gene**

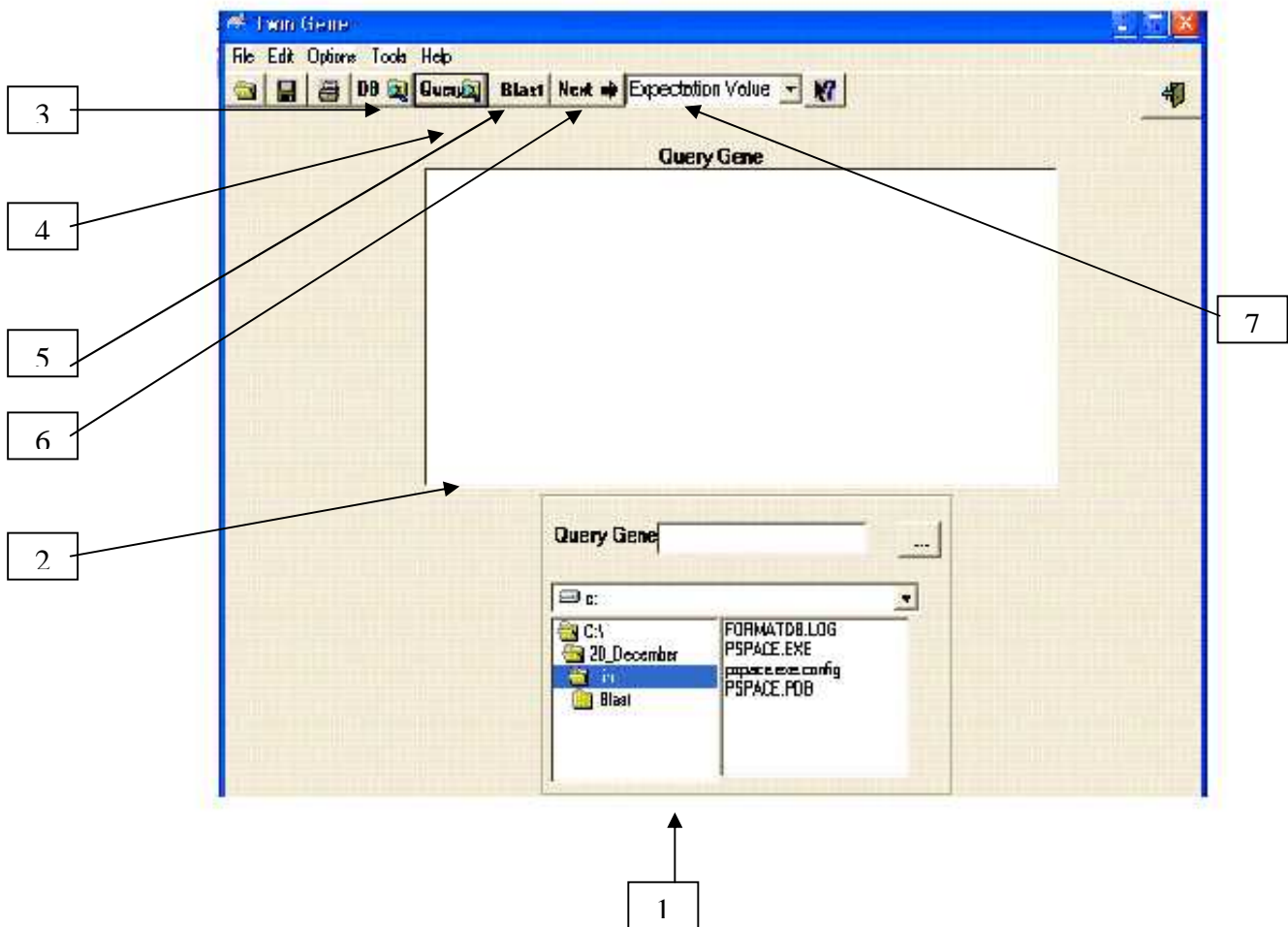






## 5.5 Interface Design

### 5.5.1 Main Window



1- Group box : an option to select the query gene file , implemented by a combination of drive list ,directory list and file list boxes .

2- Query gene : display the selected query gene .

This also can be used to paste a copied query sequence from any media , this flexibility comes from storing this text field contents in a file which is used in further processing.

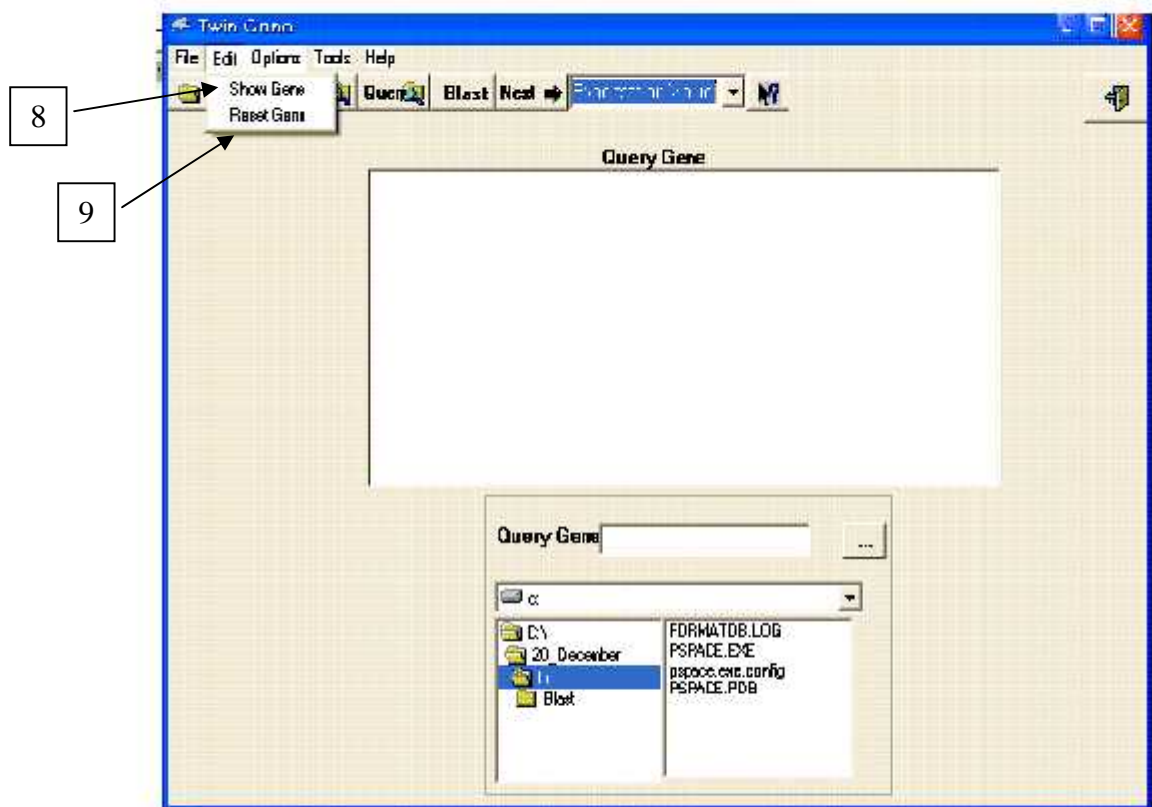
3- DB Browse : browse the database file from local computer .  
It is a replacement of writing the path of DB file manually.

4- Query : to browse the query file instead of writing its path manually.

5- Blast : run the BLAST program which receive Database and Query files as inputs and results in a (Test .out) output file .

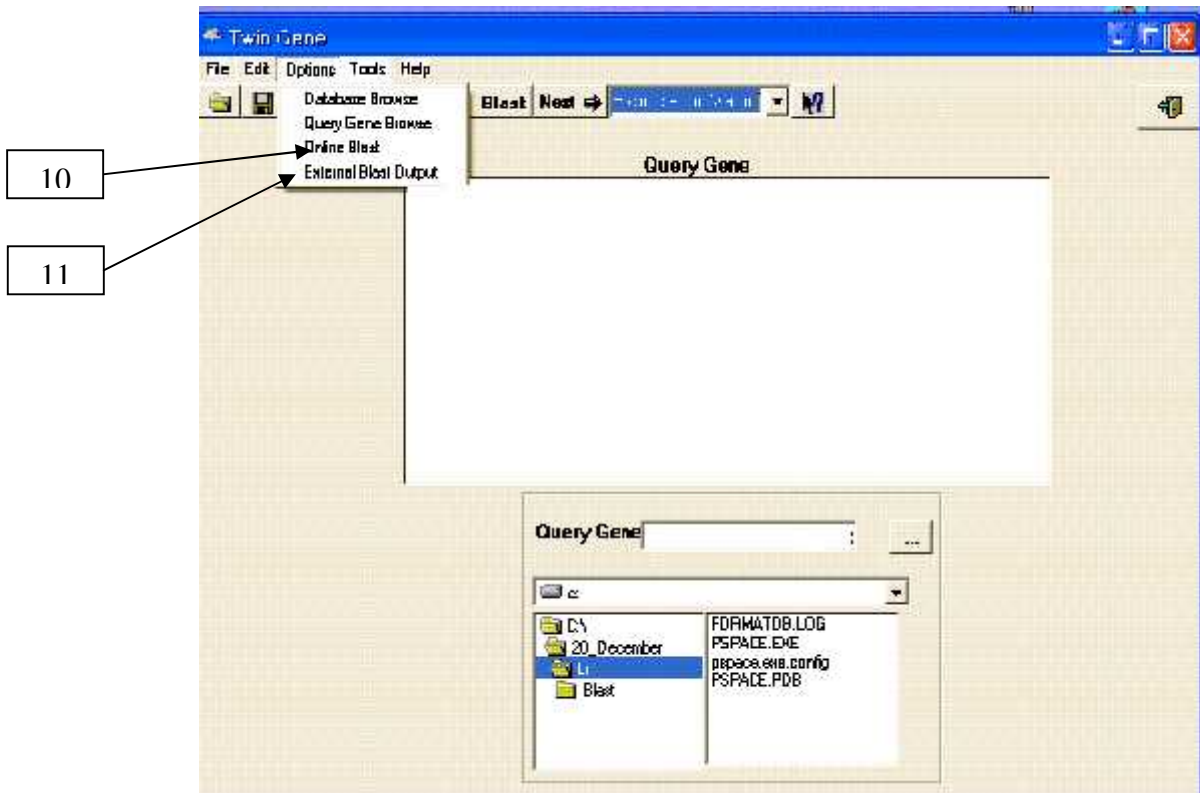
6- Next : go to Multiple Alignment page .

7- Drop down list : to select a value of expectation which is input to BLAST program .



8- Show Gene: display selected query gene on the query gene text field on screen

9- Reset Gene: clears the query gene text field on screen

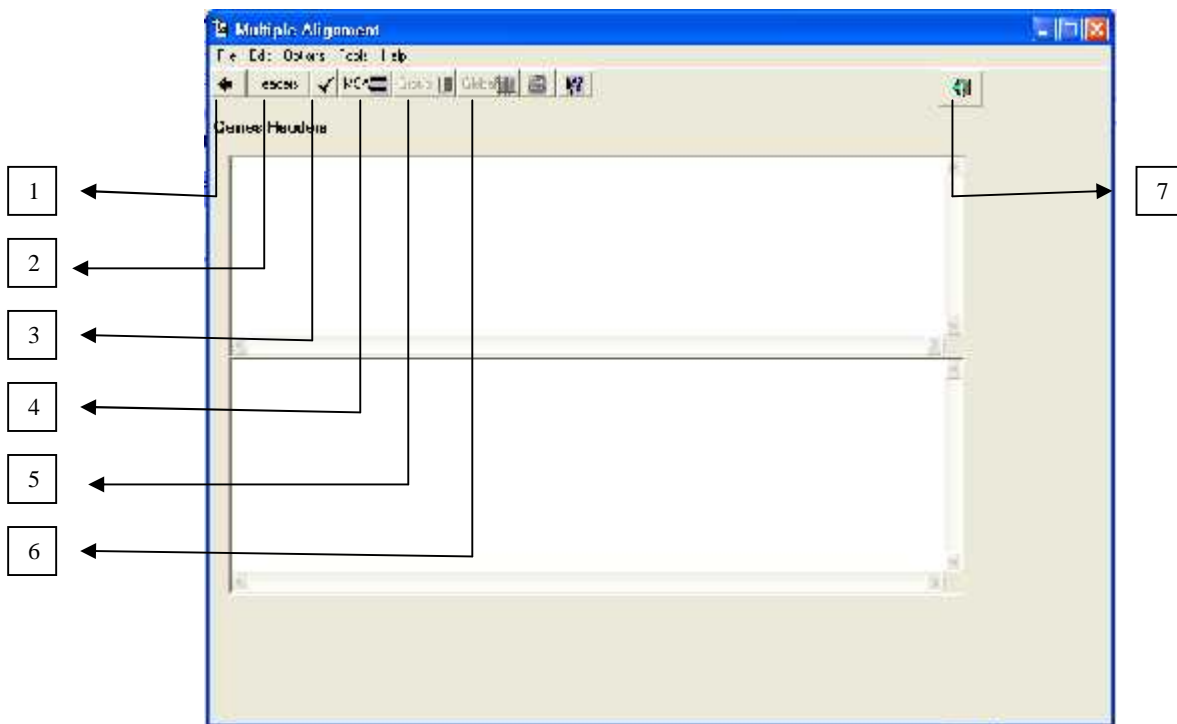


10 – Online blast : Browse the NCBI website .

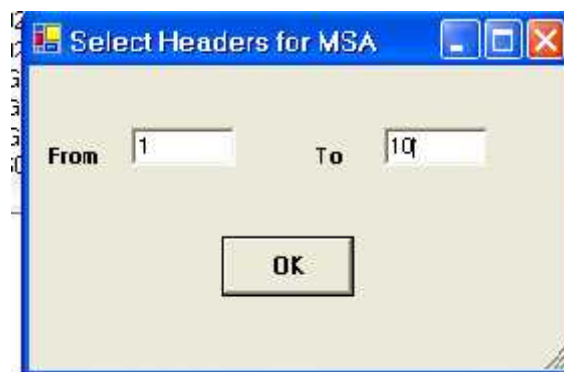
11- External blast output : imports blast results which out from previously done blast process .

## 5.5.2 Multiple Alignment window

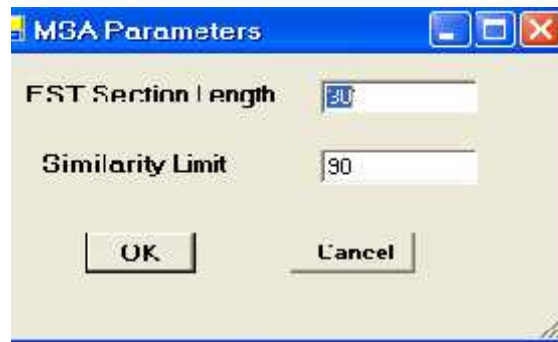
- **Toolbar and buttons in MSA form**



- 1- Go back to the main window.
- 2- Display all headers from blast output on a list .
- 3- Select number of headers for MSA by entering the first and the last indices .



4- Running the Multiple Sequence Alignment process after determining the parameters (EST section Length , Similarity Limit), then display the output on a text field.



5- Grouping

Grouping Parameters



## 6- Global Alignment

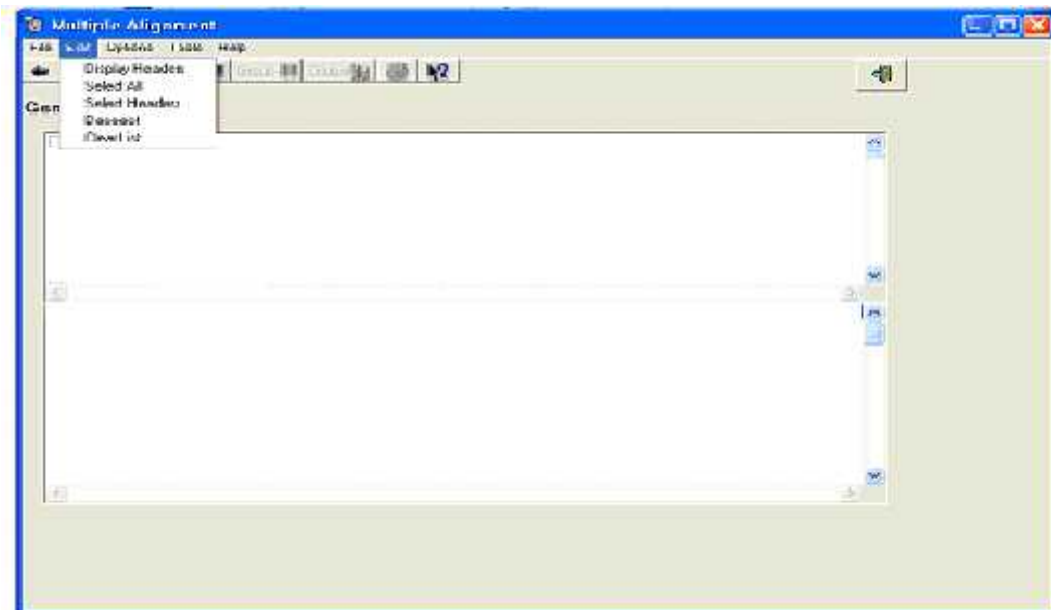


7- To exit from the program.



- **Menus in MSA form**

1- Edit menu

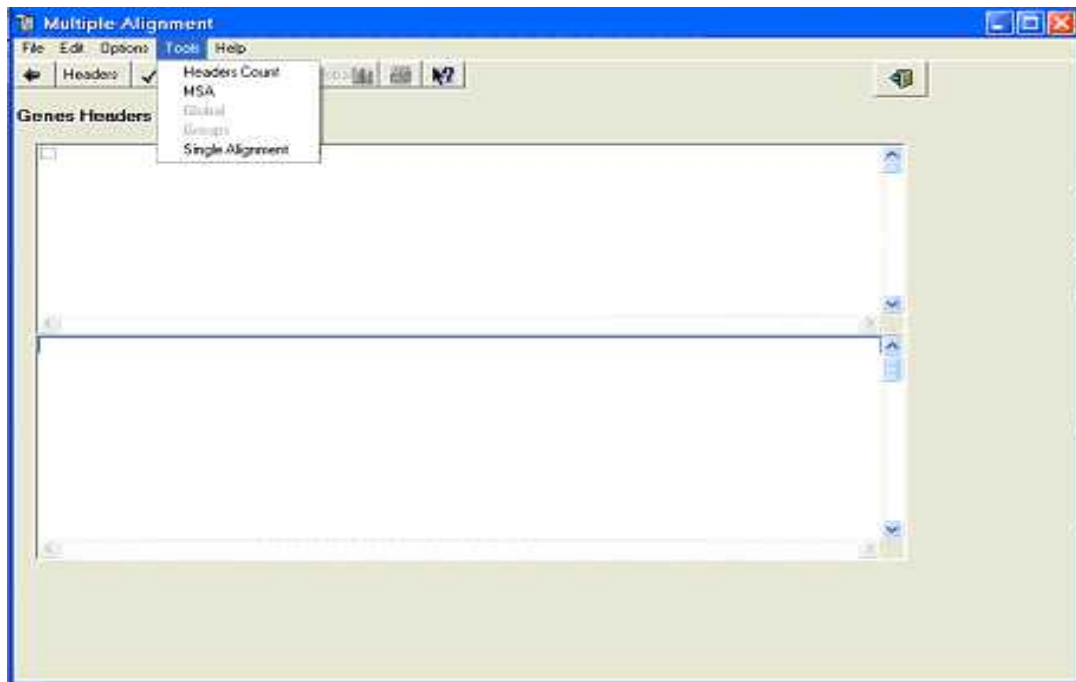


- Display Headers: Display all headers from blast output on a list .
- Select All: Select all the headers from the list for MSA process.
- Select Headers: Select number of headers for MSA by entering the first and the last indices (Like check button).
- Deselect: deselect the selected headers.
- Clear List: delete all data in the headers list.

2-Option menu



### 3-Tool menu



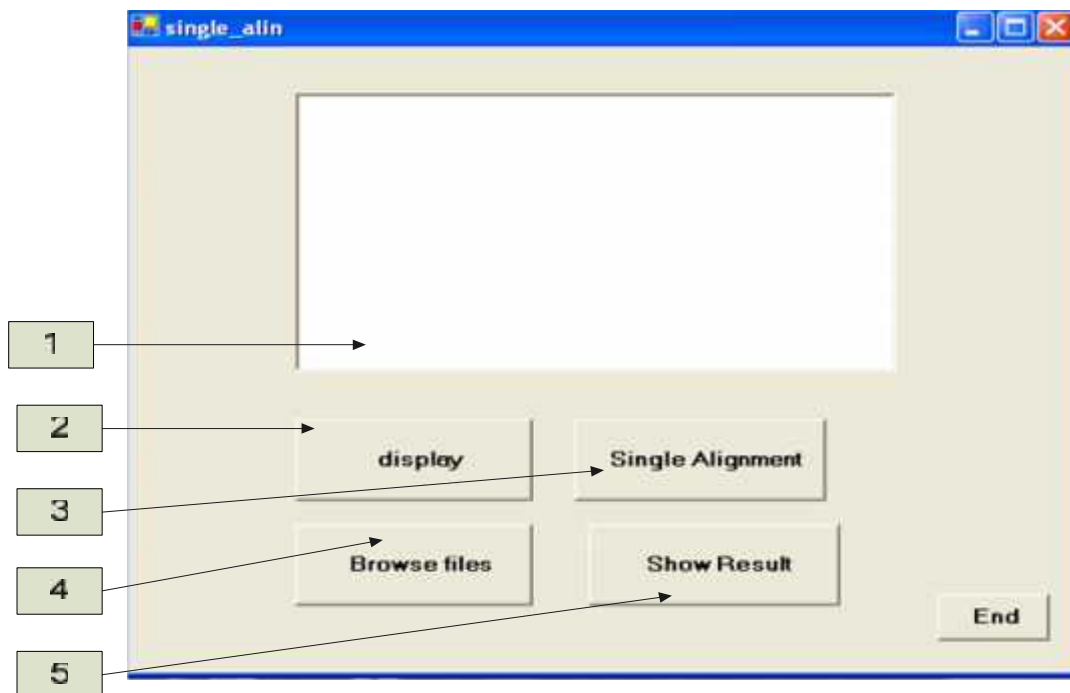
-Headers Count: calculate the number of headers shown in the list.



-MSA :perform Multiple Sequence Alignment process after entering the parameters and then show the results.



## -Single Alignment



1. Header's field: The headers that resulted from blast needed to make single alignment.
2. Display headers: this will display the headers from blast output, to enable user to select one of them to make single alignment with query.
3. Single Alignment: perform the single alignment process.
4. Browse Files: Enable the user to browse any needed file.
5. Show Result: Display the output of the single alignment.

# Chapter Six

## Implementation

### 6.1 Overview

The implementation of the project involves a number of steps or phases some of which are performed serially others in parallel, however to implement these functions , some software tools that must be installed on the server are needed such as Windows XP, Microsoft Office 2000 and Visual Basic.Net Language. This chapter describes the implementation phases of the project.

This chapter discusses each system part separately and its interaction with other parts.

### 6.2 Building the System

The basic components needed for the system are the following:

1. Operating system: Windows XP.
2. Programming languages: Visual Basic.Net.

### 6.3 Prepare the Platform of the System

Before operating the system, a number of configuration steps must be performed to prepare the suitable platform to operate the system properly.

#### 6.3.1 Install and Configure Windows XP professional

Windows XP may be installed to get more efficiency when Visual Basic.NET is needed to be installed .

### 6.3.2 Install Visual Basic.NET

During development stage The Visual Basic.NET and its framework 1.1 must be installed on the computer in order to test the previous system and develop the current one with new requirements. It is preferred to use the last version of this software (Visual Basic.NET 2003) to get more effective performance, in this project Visual Basic.NET 2002 is used .

### 6.3.3 Install BLAST Program

This program gives the similar EST'S for the input sequence in a file without alignment, there are three steps needed to setup the Standalone BLAST executable program which are stated in the previous documentation .

For testing we recommend using relatively small database like ecoli.nt.Z (1349 Kb). This is a FASTA formatted file of nucleotide sequences. Before using the BLAST program one need to format the database using the 'formatdb' program which comes with the standalone executable BLAST. Next a test nucleotide file containing the query sequence is created to run against the database; make a text file for example called test.txt with the following format :

```
>Test
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAGAGT
GTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTTAAATTTTATTGAC
TTAGGTCACATAAATACTTTAACCAATATAGGCATAGCGCACAGACAGATAAAAATTA
CAGAGTACACAACATCCATGAAACGCATTAGCACCACC
ATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAA
ACACAGAAAAAAGCCCGCACCTGACAGTGCGGGCTTTTTTTTTTCGACCAAAGGTAAC
GAGGTAACAACCATGCGAGTGTTGAAGTTCGGCGGTACATCAGTGGCAAATGCAGA
ACGTTTTCTGCGTGTTGCCGATATTCTGGAAAGCAATGCCAGGCAGGGGCAGGTGGC
CACCGTCCTCTCTGCCCCGCCAAAATCACCAACCACCTGGTGGCGATGATTGAAAA
AACCATTAGCGGCCAGGATGCTTTACCCAATATCAGCGATGCCGAACGTATTTTTGC
CGAACTTTT
```

BLAST program generates an output file called test.out stored in the Standalone BLAST directory. This file is by the system to implement its processes .[3]

## 6.4 Multiple Alignment Implementation

A multiple alignment arranges a set of sequences in a scheme where positions believed to be homologous are written in a common column. The algorithm of multiple alignment includes several parts; writing the query in one line, align the query with each EST in the BLAST output based on alignment cases which was solved by the previous system , putting four columns beside each EST (sequential number, EST direction, similarity value, and standard gene number) and coloring the EST's bases with different colors for more clarity . The output was written to an HTML file using HTML commands which breaks down the output displaying process for browser usage . This is solved by replacing the HTML file with a rich text box and coloring each base separately which takes time but not as long as it was previously and a progress watcher is used to make sure that Multiple Alignment process is performed properly . The code for Multiple Alignment is associated with the previous project document and here is the output display implementation code :

```
Private Sub MSA_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles MSA.Click
```

```
    ***** The code of Multiple Alignment process *****
```

```
    ' Set Minimum to 1 to represent the first EST being aligned.
```

```
    progressbar1.Minimum = 1
```

```
    ' Set Maximum to the total number of EST's to align.
```

```
    progressbar1.Maximum = clist.CheckedItems.Count
```

```
    ' Set the initial value of the ProgressBar.
```

```
    progressbar1.Value = 1
```

```
    ' Set the Step property to a value of 1 to represent each EST being aligned.
```

```
    progressbar1.Step = 1
```

*'\*\*\*\*\* Writing the query sequence to the rich text box \*\*\*\*\*'*

```
RichTextBox1.AppendText(Chr(13))
```

```
For J = 1 To Len(finalstt)
```

```
  If Mid(finalstt, J, 1) = "A" Then
```

```
    RichTextBox1.SelectionColor = System.Drawing.Color.Blue
```

```
    RichTextBox1.SelectedText = "A"
```

```
  ElseIf Mid(finalstt, J, 1) = "C" Then
```

```
    RichTextBox1.SelectionColor = System.Drawing.Color.Red
```

```
    RichTextBox1.SelectedText = "C"
```

```
  ElseIf Mid(finalstt, J, 1) = "T" Then
```

```
    RichTextBox1.SelectionColor = System.Drawing.Color.DarkOrange
```

```
    RichTextBox1.SelectedText = "T"
```

```
  ElseIf Mid(finalstt, J, 1) = "G" Then
```

```
    RichTextBox1.SelectionColor = System.Drawing.Color.Green
```

```
    RichTextBox1.SelectedText = "G"
```

```
  ElseIf Mid(finalstt, J, 1) = " " Then
```

```
    RichTextBox1.SelectedText = " "
```

```
  Else
```

```
    co = Mid(finalstt, J, 1)
```

```
    RichTextBox1.SelectionColor = System.Drawing.Color.Black
```

```
    RichTextBox1.SelectedText = co
```

```

        End If
    Next
    progressbar1.PerformStep()

    RichTextBox1.AppendText(Chr(13))

    '***** Writing each EST sequence to the rich text box *****

    For i = 0 To (clist.Items.Count) - 1
        If clist.GetItemChecked(i) Then
            For J = 1 To Len(stfinal(i))
                If Mid(stfinal(i), J, 1) = Mid(finalstt, J, 1) And Mid(stfinal(i), J, 1) = "A" Then
                    RichTextBox1.SelectionColor = System.Drawing.Color.Blue
                    RichTextBox1.SelectedText = "A"

                ElseIf Mid(stfinal(i), J, 1) = Mid(finalstt, J, 1) And Mid(stfinal(i), J, 1) = "C" Then
                    RichTextBox1.SelectionColor = System.Drawing.Color.Red
                    RichTextBox1.SelectedText = "C"

                ElseIf Mid(stfinal(i), J, 1) = Mid(finalstt, J, 1) And Mid(stfinal(i), J, 1) = "T" Then
                    RichTextBox1.SelectionColor=System.Drawing.Color.Orange
                    RichTextBox1.SelectedText = "T"

                ElseIf Mid(stfinal(i), J, 1) = Mid(finalstt, J, 1) And Mid(stfinal(i), J, 1) = "G" Then
                    RichTextBox1.SelectionColor = System.Drawing.Color.Green
                    RichTextBox1.SelectedText = "G"

                ElseIf Mid(stfinal(i), J, 1) = " " Then

```

```
RichTextBox1.SelectedText = " "

ElseIf Mid(stfinal(i), J, 1) = "-" And J = Len(stfinal(i)) Then
    Exit For
Else
    co = Mid(stfinal(i), J, 1)
    RichTextBox1.SelectionColor = System.Drawing.Color.Black
    RichTextBox1.SelectedText = co

End If
Next

RichTextBox1.AppendText(Chr(13))
progressbar1.PerformStep()

End If
Next

End Sub
```

## 6.5 Single Alignment Implementation

A single alignment arranges a single sequence in a scheme where the position believed to be homologous is written in a common column. The single alignment works the same way as multiple alignments except that it aligns the query gene with the chosen gene from the BLAST output. The chosen gene can be selected from a list . In single alignment , the four columns of EST information beside each EST are not appended . The single alignment output was in HTML file which takes time to be clear to the user , for that , a rich text box is used to display the output more effectively and clearly . The code for Single Alignment is associated with the previous project document and here is the output display implementation code :

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    f2.RichTextBox1.Clear()
    f2.RichTextBox1.Dock = DockStyle.Fill
    f2.RichTextBox1.Visible = True
    f2.RichTextBox1.AppendText(Chr(13))
    f2.RichTextBox1.Font = New System.Drawing.Font("Courier New", 8,
FontStyle.Regular)
```

```
Dim j As Integer
```

```
For j = 1 To Len(co7)
```

```
    If Mid(co7, j, 1) = "A" Then
```

```
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.Blue
```

```
        f2.RichTextBox1.SelectedText = "A"
```

```
    ElseIf Mid(co7, j, 1) = "C" Then
```

```
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.Red
```

```
        f2.RichTextBox1.SelectedText = "C"
```

```
    ElseIf Mid(co7, j, 1) = "T" Then
```

```
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.DarkOrange
```

```
        f2.RichTextBox1.SelectedText = "T"
```

```
    ElseIf Mid(co7, j, 1) = "G" Then
```

```
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.Green
```

```
        f2.RichTextBox1.SelectedText = "G"
```

```
    ElseIf Mid(co7, j, 1) = " " Then
```

```
        f2.RichTextBox1.SelectedText = " "
```

```
    Else
```

```
        co = Mid(co7, j, 1)
```

```
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.Black
```

```
        f2.RichTextBox1.SelectedText = co
```

```
    End If
```



```

Dim i As Integer
For j = 1 To Len(co8)

    If Mid(co8, j, 1) = Mid(co7, j, 1) And Mid(co8, j, 1) = "A" Then
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.Blue
        f2.RichTextBox1.SelectedText = "A"
    ElseIf Mid(co8, j, 1) = Mid(co7, j, 1) And Mid(co8, j, 1) = "C" Then
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.Red
        f2.RichTextBox1.SelectedText = "C"

    ElseIf Mid(co8, j, 1) = Mid(co7, j, 1) And Mid(co8, j, 1) = "T" Then
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.DarkOrange
        f2.RichTextBox1.SelectedText = "T"

    ElseIf Mid(co8, j, 1) = Mid(co7, j, 1) And Mid(co8, j, 1) = "G" Then
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.Green
        f2.RichTextBox1.SelectedText = "G"

    ElseIf Mid(co8, j, 1) = " " Then

        f2.RichTextBox1.SelectedText = " "

    Else
        co = Mid(co8, j, 1)
        f2.RichTextBox1.SelectionColor = System.Drawing.Color.Black
        f2.RichTextBox1.SelectedText = co

    End If

Next

f2.RichTextBox1.AppendText(Chr(13))
f2.Show()

End Sub

```

## 6.6 Global Alignment Implementation

A global alignment expresses the results of multiple alignment process in a general view . So that the user of the system can observe the alignment results globally and the EST's relations with query and each others based on the EST similarity and length described by the score value .

The algorithm of global alignment includes several parts; drawing the query in a special colored block , drawing aligned EST's blocks under the query block using different colors depending on the score value of each EST which is read from BLAST output file and on the parallel similar EST sections with the query from the Multiple Alignment output . Four columns are written beside each EST (sequential number, EST direction, similarity value, and standard gene number) to give introductive information of each EST block .

The code for Global Alignment is as follow :

```
FileOpen(1, Application.StartupPath & "\blast\test.out", OpenMode.Input)
line = LineInput(1)

While Not EOF(1) And Mid(line, 10, 9) <> "producing"

    If Microsoft.VisualBasic.Left(line, 9) = "Sequences" Then
        line = LineInput(1)
        f = f + 3
        Exit While
    End If
    f = f + 1
    line = LineInput(1)
End While

l = f
While l - f < clist.Items.Count And Not EOF(1)
    line = LineInput(1)
    If clist.GetItemChecked(l - f) Then
        'headers(sc) = clist.CheckedItems(l - f)
        score(sc) = Mid(line, 69, 5)
        'MsgBox(score(sc))
        sc = sc + 1
    End If
    l = l + 1
End While
FileClose(1)
```

```
While (est < RichTextBox1.Lines.Length - 1)
```

```
    lines = ""
```

```
    lines = RichTextBox1.Lines(est)
```

```
    For i = j To s + j - 1
```

```
        If lines.Chars(i) = "C" And i = s + j - 1 Or lines.Chars(i) = "A" And i = s + j - 1 Or  
lines.Chars(i) = "T" And i = s + j - 1 Or lines.Chars(i) = "G" And i = s + j - 1 Then
```

```
            startend(est - 3)(k) = ((i - j) * fs / s) + 70
```

```
            k = k + 1
```

```
        Exit For
```

```
    ElseIf lines.Chars(i) = "-" And i = s + j - 1 Then
```

```
        Exit For
```

```
    ElseIf lines.Chars(i) = "C" And lines.Chars(1 + i) = "-" Or lines.Chars(i) = "C" And  
lines.Chars(1 + i) = " " Or lines.Chars(1 + i) = " " And lines.Chars(i) = "A" Or lines.Chars(i) =  
"A" And lines.Chars(1 + i) = "-" Or lines.Chars(1 + i) = " " And lines.Chars(i) = "G" Or  
lines.Chars(i) = "G" And lines.Chars(1 + i) = "-" Or lines.Chars(1 + i) = " " And lines.Chars(i)  
= "T" Or lines.Chars(i) = "T" And lines.Chars(1 + i) = "-" Then
```

```
        startend(est - 3)(k) = ((i - j) * fs / s) + 70
```

```
        k = k + 1
```

```
    ElseIf lines.Chars(i) = "C" And i = j Or lines.Chars(i) = "A" And i = j Or  
lines.Chars(i) = "T" And i = j Or lines.Chars(i) = "G" And i = j Then
```

```
        startend(est - 3)(k) = 70
```

```
        k = k + 1
```

```
    ElseIf lines.Chars(i) = "-" And lines.Chars(1 + i) = "C" Or lines.Chars(i) = " " And  
lines.Chars(1 + i) = "C" Or lines.Chars(i) = "-" And lines.Chars(1 + i) = "T" Or lines.Chars(i) =  
" " And lines.Chars(1 + i) = "T" Or lines.Chars(1 + i) = "G" And lines.Chars(i) = " " Or  
lines.Chars(1 + i) = "G" And lines.Chars(i) = "-" Or lines.Chars(i) = "-" And lines.Chars(1 + i)  
= "A" Or lines.Chars(i) = " " And lines.Chars(1 + i) = "A" Then
```

```
        startend(est - 3)(k) = ((i + 1 - j) * fs / s) + 70
```

```
        k = k + 1
```

```
    End If
```

```
Next
```

```

' The colors of EST's
If score(est - 3) >= 0 And score(est - 3) <= 40 Then
    c(est - 3) = Color.Black
ElseIf score(est - 3) >= 41 And score(est - 3) <= 50 Then
    c(est - 3) = Color.Yellow
ElseIf score(est - 3) >= 51 And score(est - 3) <= 80 Then
    c(est - 3) = Color.Lime
ElseIf score(est - 3) >= 81 And score(est - 3) <= 200 Then
    c(est - 3) = Color.Fuchsia
ElseIf score(est - 3) >= 201 And score(est - 3) <= 5000 Then
    c(est - 3) = Color.Red
End If

```

```

'The numbers and directions of EST's
f4.number(est - 3) = Mid(lines, 1, 3)
If lines.Chars(8) = "<" Then
    f4.arrow(est - 3) = "<--"
Else
    f4.arrow(est - 3) = "-->"
End If

```

```

est = est + 1
k = 0
End While

```

Protected Overrides Sub OnPaint(ByVal paintEvent As PaintEventArgs)

```

g.DrawString("Query", drawFont1, drawBrush1, 0, lb + 2)
g.FillRectangle(brush, 70, lb + 2, Me.Size.Width - 100, 10) '82

For est = 0 To sted.GetUpperBound(0)
    For estno = 0 To sted(est).GetUpperBound(0) - 1 Step 2
        brush = New SolidBrush(colors(est))
        g.FillRectangle(brush, sted(est)(estno), (est * 20) + lb + 22, sted(est)(estno + 1)
- sted(est)(estno), 10)
        Next
    Next

    Dim drawFont As New Font("courier new", 10)
    Dim drawBrush As New SolidBrush(Color.Black)
    g.DrawString(number(est) & " " & arrow(est), drawFont, drawBrush, 0, (est * 20)
+ lb + 22)
    ' g.DrawString(h(est), drawFont, drawBrush, 70, (est * 20) + lb + 22)
    Next
    Button1.Location = New Point(500, (est * 20) + lb + 50)

End Sub ' OnPaint

```

## 6.7 Grouping Implementation

In this phase the paralogous sequence variants (PSV) and Twin genes are determined . To classify the genes resulted from multiple alignment into groups , an array is built to store the positions of differences of each EST with the query. Then each EST is compared with all other EST's. If two EST's agree on difference greater than (variable\* length of the EST) then these two EST's are classified in the same group , otherwise , they are in the first group (Query gene group) . The group is displayed by writing its EST's and the composed PSV on a rich text box and each PSV is translated to its analogous protein .

The code for grouping is as follows:

```
Private Sub groups_Click(ByVal sender As .... ..)

    stt = stt.Remove(Len(stt) - 1, 1)

    f7.RichTextBox1.Text = "PSV1"

    f7.RichTextBox1.AppendText(Chr(13))
    f7.RichTextBox1.AppendText(stt)
    f7.RichTextBox1.AppendText(Chr(13))

    For i = 0 To no
        For t = 0 To Len(stt)
            dif2(i, t) = " "
        Next
    Next

    For i = 0 To no
        For n = 0 To Len(st(i)) - 1
            If st(i).Chars(n) = "C" Or st(i).Chars(n) = "T" Or st(i).Chars(n) = "G" Or st(i).Chars(n)
= "A" Then
                Exit For
            End If
        Next
    Next
End Sub
```

```

    End If
Next
For t = 0 To n - 1 + imstart
    dif(i, t) = " "

Next

For j = (n + imstart) To (Len(st(i)) - imstart)
    If Mid(st(i), j, 1) <> Mid(stt, j, 1) Then
        dif(i, j) = Mid(st(i), j, 1) ' a row of differences positions for the est

    Else
        dif(i, j) = " "

    End If

Next

For t = (Len(st(i)) - imstart + 1) To Len(st(i)) - 1
    dif(i, t) = " "
Next
Next

psvgroup(0).psv = 1
maxgp = 1 ' at least one group
For i = 0 To (clist.Items.Count) - 1
    cnt = 0
    For j = 0 To (clist.Items.Count) - 1 ' compare each est with all est's
        cnt = 0
        If clist.GetItemChecked(i) And clist.GetItemChecked(j) Then

```

```

If i <> j Then
  For x = 1 To Len(st(j))
    If dif(j, x - 1) <> " " And dif(j, x - 1) = dif(i, x - 1) Then
      cnt += 1 ' # of differences
      If cnt > (minimum_percent * totlen(i) / 100) Then
        dif2(j, x - 1) = dif(i, x - 1)
      End If
    End If
  Next
End If

maxgp = Max(maxgp, psvgroup(i).psv)
If j > 0 And i <> j Then
  If cnt > (minimum_percent * totlen(i) / 100) Then

    If psvgroup(i).psv = 0 And psvgroup(j).psv = 0 Then
      psvgroup(i).psv = maxgp + 1 ' the two est's(inner and outer don't
      psvgroup(j).psv = maxgp + 1 ' have a group; start new group

    ElseIf psvgroup(i).psv = 0 And psvgroup(j).psv <> 0 Then ' the outer est hasnt
a group
      psvgroup(i).psv = psvgroup(j).psv ' give this group to the outer

    ElseIf psvgroup(i).psv <> 0 And psvgroup(j).psv = 0 Then ' the outer est has a
group
      psvgroup(j).psv = psvgroup(i).psv ' give this group to the inner est

    End If
  End If
End If
End If

```

```

        End If
    Next
Next

g = 0
For i = 0 To (clist.Items.Count) - 1
    If clist.GetItemChecked(i) And (idaverage(no) > similarity_limit) Then
        If psvgroup(i).psv = 0 Then
            psvgroup(i).psv = 1
        End If

        If (psvgroup(i).psv = 1) Then
            f7.RichTextBox1.AppendText(Chr(13))
            f7.RichTextBox1.SelectionColor = System.Drawing.Color.Blue
            f7.RichTextBox1.SelectedText = i + 1 & " "
            f7.RichTextBox1.SelectionColor = System.Drawing.Color.Black
            f7.RichTextBox1.SelectedText = st(i)
            f7.RichTextBox1.AppendText(Chr(13))
        End If

        g = Max(g, psvgroup(i).psv)
    End If
Next

f7.Text = "There are " & g & "groups !"
' f7.RichTextBox1.AppendText(Chr(13))
f7.RichTextBox1.AppendText(Chr(13))

f7.RichTextBox1.AppendText("*****")
f7.RichTextBox1.AppendText(Chr(13))

```



```

For p = 2 To g
  query = stt

  f7.RichTextBox1.AppendText("PSV" & p)
  f7.RichTextBox1.AppendText(Chr(13))

  For i = 0 To clist.Items.Count - 1
    If psvgroup(i).psv = p Then
      For x = 1 To Len(st(i))

        If dif2(i, x - 1) <> " " And dif2(i, x - 1) <> "-" And dif2(i, x - 1) <> "N" Then
          Mid(query, x - 1, 1) = dif2(i, x - 1)
        End If
      Next
    End If
  Next

  For t = 0 To Len(stt) - 1
    If query.Chars(t) = stt.Chars(t) Then
      f7.RichTextBox1.SelectionColor = System.Drawing.Color.Black
      f7.RichTextBox1.SelectedText = query.Chars(t)
    Else
      f7.RichTextBox1.SelectionColor = System.Drawing.Color.Red
      f7.RichTextBox1.SelectedText = query.Chars(t)
    End If
  Next

  f7.RichTextBox1.AppendText(Chr(13))

  For j = 0 To clist.Items.Count - 1

```

```
    If psvgroup(j).psv = p Then
        f7.RichTextBox1.SelectionColor = System.Drawing.Color.Blue
        f7.RichTextBox1.SelectedText = j + 1 & " "
        f7.RichTextBox1.SelectionColor = System.Drawing.Color.Black
        f7.RichTextBox1.SelectedText = st(j)
        f7.RichTextBox1.AppendText(Chr(13))
    End If
Next

f7.RichTextBox1.AppendText("*****")
    f7.RichTextBox1.AppendText(Chr(13))
Next

End Sub
```

# Chapter Seven

## System Testing

### 7.1 Overview

Testing is one of the significant steps of software engineering. As an initial testing has been performed on the previous system to discover the development aspects, this testing is done to the current system. In parallel with the implementation phase and at the end with testing we can put the system after all the work has been done under spot to see if it is good as the expectations, and to find out the mistakes, problems or other unexpected emergent and also to see if the system fulfills the requirement.

### 7.2 Testing Procedures

#### 7.2.1 Unit Testing

At this stage, individual components are tested to ensure the correctness of its functionality. Each component should be tested independently.

The tested components are the following:

- Import query gene
- Import formatted database file
- Display headers from BLAST results
- Display the aligned query section with each EST
- Select headers for multiple alignment
- Multiple alignment result display
- Read multiple alignment output
- Read score from BLAST output
- Draw colored bars for each EST
- Classifying EST's into groups
- Twin Gene composing
- Protein translation
- Grouping display
- Single alignment display

After testing the previous items, system had outputs the expected values and Unit Testing is Passed Successfully.

### **7.2.2 Module Testing**

At this stage, we collected the related components from the unit testing stage in groups, each group called module. A module encapsulates related components so it can be tested without other components.

After testing these collections of related units (Modules), we tested each module as a whole. For each module, a number of parameters entered to its units and results are checked.

Example of Modules that are tested correctly

- Input module
- Multiple Sequence Alignment module
- Global Alignment module
- Single Alignment Module
- Grouping module

This type of testing is implemented using the Black-Box strategy. This type of testing focuses on studying the system as a black box whose behavior can only be determined by studying its inputs and related outputs. This testing is only concerned with the functionality and not the implementation of software.

The following is an example of testing parameter inputs in the Multiple Sequence Alignment module:

<b>Inputs</b>	<b>Expected Values</b>	<b>Actual Values</b>
EST section length= -30 Similarity limit=90	The system won't continue the execution. It will give an error message	The system doesn't continue the execution. It gives an error message
EST section length= 0 Similarity limit=90	The system will continue the execution.	The system continues the execution.
EST section length= 30 Similarity limit=90	The system will continue the execution.	The system continues the execution.
EST section length= 30 Similarity limit=120	The system won't continue the execution. It will give an error message	The system doesn't continue the execution. It gives an error message
EST section length= 30 Similarity limit= -100	The system won't continue the execution. It will give an error message	The system doesn't continue the execution. It gives an error message
EST section length= 0 Similarity limit=120	The system won't continue the execution. It will give an error message	The system doesn't continue the execution. It gives an error message
EST section length= 0 Similarity limit= -100	The system won't continue the execution. It will give an error message	The system doesn't continue the execution. It gives an error message
EST section length= -30 Similarity limit=120	The system won't continue the execution. It will give an error message	The system doesn't continue the execution. It gives an error message
EST section length= -30 Similarity limit= -100	The system won't continue the execution. It will give an error message	The system doesn't continue the execution. It gives an error message

### **7.2.3 Sub-Systems Testing**

Here, the testing performed over a collection of modules which was integrated in subsystems.

The tested subsystems are the following:

- Input subsystem.
- Multiple Alignment subsystem
- Single Alignment subsystem
- Global Alignment subsystem
- Grouping subsystem

### **7.2.4 System Testing**

Sub-systems are integrated to constitute the whole system . System is tested with a complete process of import the query gene and database -> execute the BLAST program -> displaying headers and alignments -> select headers for MA -> perform MA -> perform grouping -> perform global alignment .

Real data are filled through that sequence and traced to find that results are as expected so that the system processes work properly. System testing is also accomplished through transferring the system on different platforms.

# **Chapter Eight**

## **Deployment and Maintenance**

### **8.1 Overview**

This chapter describes the establishment of the environment that the system will work in, process of deployment and the maintenance plan.

### **8.2 Establishment of production Environment**

In order to install this system (product) a certain environment for the user to be established. To operate the system a PC with an operating system is necessary .

- Install the operating system .
- Install the .Net framework .
- Install the Twin Gene program package.

### **8.3 Migration and Deployment Plan**

There are three steps required to move any windows application from the development environment to a production directory :

1. Build ,or compile the Windows application, this compilation creates a dynamic-link library (DLL) file in the \bin directory that contains all of the code for Twin Gene
2. Copy the necessary windows application files(after compilation Twin Gene) in the development directory to the production directory, which are:
  - The \bin directory and the DLL files.
  - All windows forms
  - The Execution File
  - Any additional files such as app.config and AssemblyInfo.vb

3. Package and build the production solution.

## 8.4 Implementation Plan

To start working with the system there are three techniques, and the suitable way finally is picked, the major three ways are:

- **Direct:** the old system is totally removed and the new one is used
- **Pilot:** the old system is work with new system for a period of time then the old on is removed gradually and the new one continue in working
- **Parallel:** the two systems are working together.

Twin Gene system needs the Direct technique, means that the new Twin Gene system will be installed after uninstalling the old one.

## 8.5 Installing Twin Gene System

The following steps are needed to install the program:

1. You Should Install .Net Framework

"Install .Net Framework from the same Directory you are inside"

2. Inside debug folder, Click on Setup

3. Choose where to install.

You Must Install it On C:\Twin Gene Project\  
Or D:\Twin Gene Project\  
Or Any drive directly C:\ or ...





## How To Complete The MAINTENANCE LOG

This change control log form is included as a suggested format for recording and maintaining software change request data, including changes to documentation. A Detailed Status Information form is available to record supplementary details. The log and software change requests should be maintained in the Systems Project Notebook.

Log Date: \_\_\_\_/\_\_\_\_/\_\_\_\_

Page #: \_\_\_\_

System Name: _____										
Request #	Reqmnt #	Date Submitted	Approval			Status				
			Change Approved	Change Not Approved	Hold (Future Enhancement)	Technical Evaluation Phase	Change In-Progress	Cancelled	Target Date	Date Complete

Definition of the LOG fields

**Page No:** Enter the appropriate page number of the log sheet.

**Log Date:** Enter the date control log was started.

**System Name:** Enter the name and acronym of the system to be managed.

**Request No:** Enter the unique sequential number assigned to each request on the request form (i.e., software change request form, etc.)

**Reqmnt No:** Enter the unique number of the requirement to be changed (if known) on the request form.

**Date Submitted:** Enter the date the request was submitted to the maintenance team.

**Approval:** This area is for recording request approval information obtained from the request form.

**Change Approved:** Enter the date the request was approved.

**Change Not Approved:** Enter the date the request was disapproved.

**Hold (Future Enhancement):** Enter the date the request was placed on "Hold."

**Status:** This area is for recording basic information about the status of a request.

**Technical Evaluation Phase:** Enter the date the technical evaluation of the request commenced.

**Change In-Progress:** Enter the date work began on the request. Usually, the areas "Technical Evaluation Phase" (if applicable) and "Change Approved" should be entered prior to posting the "Change In-Progress" date. Work on most requests should not be initiated without a technical evaluation and formal approval in the request form.

**Canceled:** Enter the date the request was canceled.

**Target Date:** Enter the estimated date that the request will be completed and ready for release/implementation.

**Date Complete:** Enter the actual date the request was implemented.

# **Chapter Nine**

## **Conclusions, Problems and Future Works**

### **9.1 Conclusions**

There are many lessons gained through this project. Here are some conclusions described briefly:

1. The interface is improved to achieve an easy interaction .
2. Using rich text box to display the output, instead of using HTML files, increases the system performance.
3. This system can be used to classify genes into groups, to find out the Twin Gene.
4. This system is capable to translate the DNA representation of the gene into amino acids.
5. Using colored blocks to represent the Global Alignment is descriptive.
6. Finding out the Twin Gene in computerized experiments is more accurate , easier and faster than traditional ways in laboratories .

### **9.2 Problems**

There were some difficulties that hamper our progress during the development and implementation processes. We will mention the prominent problems like:

1. Since the previous system was implemented using Visual Basic .Net, we had to learn the language although it doesn't support rapid execution .

2. Since the program deals with external files and programs, it was difficult to execute it from any drive or directory .
3. Running BLAST program on line is not implemented because it is not easy to read its results from a web page .

### **9.3 Future Work**

There may be a great work to be done depending on this system. We can remember some of ideas concentrating on completing the system to the universality surname:

- Running BLAST on line and reading its results for further processing .
- Importing query genes from the web by sending their accession number .
- Solving the problem of processing very long sequence genes .

## References

### Web Sites

[1] ] [http:// Bioinformatics.ubc.ca](http://Bioinformatics.ubc.ca).

[2] [www.lectures/molgen.mpg.de/online\\_lectures.html](http://www.lectures/molgen.mpg.de/online_lectures.html).

[3] [www.genome.nih.gov](http://www.genome.nih.gov).

### Books

- Twin Gene Project documentation, by 1<sup>st</sup> edition

- Bandy Niteen, Visual Studio .NET for Dummies, Alfarook Publishing, Egypt, 2002.

- Deitel, Visual Basic Net How To Program 2 Ed, Prentice Hall , USA , 2003 .

- MSDN Training - Visual Basic.NET