A MOBILE APPLICATION ON PERFORMING BLAST ANALYSIS AND LOCAL ALIGNMENT

A DISSERTATION SUBMITTED TO BRAC UNIVERSITY IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Submitted by
Md. Ismail Kiron
Student ID: 12101076
Md. Enamul Haque Sumon
Student ID: 15341034
Jannatul Maksuda Mourey
Student ID: 14101270
Md. Zayed Ullah
Student ID: 12101119
December 2015

Department of Computer Science & Engineering
BRAC University
Dhaka, Bangladesh

Dedicated to our Parents

DECLARATION

We hereby declare that the project work embodying the results reported in this thesis entitled **"A mobile application on performing BLAST analysis and local alignment"** submitted by the undersigned has been carried out under the supervision of Dr. Md. Muhidul Islam Khan, Assistant Professor, Department of Computer Science and Engineering, BRAC University, Dhaka and Dr. Aparna Islam, Associate Professor, Biotechnology Program, Department of Mathematics and Natural Sciences, BRAC University, Dhaka. It is further declared that the project work presented here is original and has not been submitted to any other institution for any degree or diploma.

(Md. Ismail Kiron)
Candidate

(Md. Enamul Haque Sumon)
Candidate

(Jannatul Maksuda Mourey)
Candidate

(Md. Zayed Ullah)
Candidate

Certified

(Dr. Md. Muhidul Islam Khan)
Supervisor

Assistant Professor
Department of Computer Science
and Engineering
BRAC University, Dhaka

(Dr. Aparna Islam)
Co-supervisor

Associate Professor
Department of Mathematics and
Natural Sciences
BRAC University, Dhaka

ACKNOWLEDGEMENT

To begin with we would like to express our gratitude to the Almighty for His blessings that had given us the strength and patience to be able to successfully complete this dissertation.

We are very much thankful to Dr. Md. Haider Ali, Professor and Chairperson, Department of Computer Science and Engineering, BRAC University for his support and appreciation to our work.

We would like to thank our thesis supervisor Dr. Md Muhidul Islam Khan, Assistant Professor, Department of Computer Science and Engineering, BRAC University for his all-out help and cooperation during the period of our thesis. His guidance played a pivotal part in completing our thesis.

We are also very thankful to Dr. Aparna Islam, our co-supervisor, Associate Professor, Department of Mathematics and Natural Sciences, BRAC University for her immense support and help in the interpretation process of the output as well as her crucial role in writing the thesis paper.

And lastly we would like to thank our very first and core ex. supervisor, Ms. Farzana Rashid, Lecturer (Study Leave) Department of Computer Science and Engineering, BRAC University who helped us in selecting such a wonderful field and topic to work on the first place. But unfortunately we had to part as she is currently on a study leave and could not assist us any farther.

December, 2015

LIST OF ABBREVIATIONS

ADT	Android Development Tool
BLAST	Basic Local Alignment Search Tool
DNA	Deoxyribonucleic Acid
HTML	Hypertext Markup Language
JDK	Java Development Kit
JVM	Java Virtual Machine
NCBI	National Center for Biotechnology Information
PHP	Personal Home Page
XAMPP	Cross-Platform, Apache, MariaDB, PHP, Perl
XML	Extensible Markup Language

ABSTRACT

In this ever-fast running modern world of technologies, bioinformatics is not lagging behind. It is now possible to connect one's smart phone to the huge database of bioinformatics where one can find thousands of gene and organism information in a few seconds. This project of an Android mobile application is another dimension that has been added to the features of smart phones. This mobile app will allow the researchers and the students in the field of bioinformatics to have access to the NCBI website's BLAST protein sequences as well as the Blosum62 matrix used for the Smith-Waterman algorithm to identify and deduce similarities between sequences. Local alignment and global alignment features were selected for the application. Upon testing of the programming language Java was found to be the most optimal language. On the basis of time complexity, space complexity etc. Smith-Waterman and BLAST algorithms were chosen for the application. First the algorithms were implemented for the computer system using Java. This was done in two parts, firstly implementation of the algorithms and secondly parsing data from the database server. Gradually the program was incorporated into smart phone system. Finally, the application was evaluated in the smart phone system using both sequence input and accession ID input of protein sequences. The results obtained from the app in the smart phone system were compared to the existing computer system which proved that new system is providing the same results. In future this app will be expanded to other platforms of mobile operating systems and additional functions will be added.

Keywords: BLAST, Smith-Waterman, protein sequence, alignment, Android, Java.

TABLE OF CONTENTS

CHAPTER I: Introduction

1. INTRODUCTION

1.1 Biotechnology, genetics and bioinformatics

Modern technology is the wave in science. It represents an interface of basic and applied sciences with gradual and subtle transformation of science into technology. Over the past twenty five years, a mere sliver of recorded time, the world of biology and indeed the world in general has been transformed by the technical tools of a field now known as Biotechnology.

Biotechnology is defined as the application of scientific and engineering principles to the processing of materials by biological agents to provide goods and services. Since the inception of the word 'biotechnology' in 1919, this had made changes in various life science processes to improve existence. Biotechnology comprises a number of technologies based up on increasing understanding of biology at the cellular and molecular levels [1]. Biotechnology has impactful applications on a vast number of fields. Biotech has been producing innumerable new products that have the possibility to alter our lives for the betterment [2]. Genetically modified crops, transgenic animals, DNA recombinant medicines, genetically engineered microbes for waste management and also biosensors have improved our life. Biotechnology plays a big role in the biofuels industry also.

Enhancing plant and animal behavior by traditional methods like cross-pollination, grafting, and cross-breeding are time-consuming. Biotech advancement has led for specific changes to be made rapidly, on a molecular level through over-expression or removal of genes, or by introduction of foreign genes. Environmental biotechnology gives response to a chemical that helps to measure the level of damage caused or the exposure of the toxic or the pollution effect caused, so it can be regulated accordingly. Even in forensic analysis, biotechnology has brought huge positive changes. With the use of minimum amount of DNA samples from a crime scene or body it is now possible to identify the criminal or determine the percentage and pedigree. All these have been possible because of the knowledge of genetics.

Genetics has progressed rapidly in the last few decades. Genetics in general is the study of genes, characters heredity, and genetic variation in living organisms. A distinct part of genetics is genomics, which works on recombinant DNA and DNA sequencing methods. Bioinformatics assemble these sequences and analyze the function and structure of genomes. These activities completely define genomics. Increased understanding of human genetics has the potential to predict how people, depending on their precise genetic makeup, will respond to certain drugs and environment condition. Genomics along with proteomic which deals with protein structures, function and protein interaction in physiological metabolic pathways, now it is possible to develop personalized medicine. Pharmacokinetics analyze the DNA and can tell intolerance or side effects of a drug application may cause. Viewing these importance of human genome sequence New Zealand has established a databank of DNA profiles. It contains over 70,000 DNA profiles [3].

All these achievements have now become possible because of our present understanding of DNA at molecular level. For better understanding of central molecular dogma of life and implementation of biotechnology there is Bioinformatics; a discipline that has been developed to improve on methods for storing, retrieving, organizing and analyzing molecular data at DNA and protein levels [4]. One of the major activity in bioinformatics is to develop software tools to generate useful biological knowledge. The other activities include deciphering this data to understand the life system at DNA and protein levels, and also to understand the interaction of DNA and proteins and cell signaling.

1.2 Popularity of smartphones

In recent times there is a sensation that has spread around the globe expeditiously and it is smart phones. Smart phone usage is increasing due to its functionality. There are statistics that suggest that, mobile phones may someday overtake desktop computers for personal use. A study in 2009 by The Nielsen Company found that, an escalating rate of smart phones usage among American wireless subscribers is 14% at the end of 2008; 19% in Q3 2009 and 21% in Q4 2009. According to Roger Entner, Senior Vice President of Research and Insights in Nielsen’s Telecom Practice, the study findings indicated that in 2009, the United States was “at the beginning of a new wireless era where smart phones will become the standard device consumers will use to connect to friends, the Internet and the world at large” [5].

1.3 Bioinformatics in smartphones

National Center for Biotechnology Information (NCBI) is basically a database that combines biotechnology, genomics and bioinformatics and brings them to our disposal to use as we please. NCBI does not only have a GenBank, a nucleic acid sequence database but also provides analysis and retrieve resources for the data in GenBank and other biological data that is made available through the NCBI website. So NCBI is the ultimate tool for those in the biotechnology or genetics field. The preliminary data of NCBI gives the base line information to do further study in genomics, proteomics and cell signaling thus gives in some idea of how life functions. Therefore, availability of these useful primary database in smart phones will make this vast information hub at our disposal 24*7. This will benefit the researchers and scientists immensely. The database and resources of NCBI at our fingertips anytime anywhere. No one would have to wait to get in front of a personal computer if they have a query suddenly comes to mind. If such an application is made for the smart phone, it will make the entire field of bioinformatics progress faster, not to mention make life easier for the researchers.

However, implementing such an application into the smart phone has several stages. Moreover, there are several factors that need to be taken under consideration. For example which available algorithm would be best suited for the smart phone systems? More importantly, how it will affect the time complexity and the space complexity of the operations and databases. Finally, efficiency of the data retrieval and comparison with the presently used mode at personal computer. Thus, this paper will elaborate on these factors and try to produce an optimal application for the smart phones.

1.4 Review on programming languages for bioinformatics

"Bioinfo7rmatic analyses involve a range of tasks and processes. Diverse programs have been written for various bioinformatics applications using every available language. Because of the size of bioinformatics datasets, computation time is not trivial, and efficiencies in computational speed are desirable" [6]. There are a range of programming languages to choose from. Some of them are C, C++, C#, Java, Perl, and Python. The mentioned languages can be divided into three groups. Perl and Python are the script group, Java and C# in semi-compiled group and C and C++ in the compiled group [6]. These languages are broadly used by many programmers to code BLAST algorithms in bioinformatics. Yet some of these languages performs quite differently from each other. C and C++ gives the fast performance among the languages that was mentioned earlier. On the other hand Perl, and Python are better than everyone else in terms of performances. "Java and C# appeared to be a compromise between the flexibility of Perl and Python and the fast performance of C and C++. The relative performance of the tested languages did not change from Windows to Linux and no clear evidence of a faster operating system was found" [6].

1.5 Objectives

Biotechnology and bioinformatics has advanced very significantly in the last decade. So biological data are being produced at a phenomenal rate. Due to this vast nature of data, it is not only big challenge for biology but also challenge for computation.

Unfortunately, still there is no application made based on the smart phone system to provide assistance to the scientific community. Therefore, the goal of the current study is to consider the factors and provide the best possible application for the bioinformatics researchers. This thesis is mainly based on developing a mobile application which will perform BLAST analysis by providing protein sequence or accession ID and also local alignment of two given protein sequences. To make such an application, all the available algorithms would have to be considered and tested to find the best suited algorithm for the application, according to the smartphone platform and user requirements. A proper database would also have to be selected among the available ones. To implement the local and global alignment algorithms, the existing programming languages would have to be compared to find the best suited one for the smartphone application. Among them accessible web designing tools the best one would have to be implemented to connect the smartphone application to the database server. Finally a user friendly interface would have to be implemented to ensure comfortable use by consumers.

CHAPTER II: Materials and Tools

2. MATERIALS AND TOOLS

2.1 Materials

2.1.1 Java

Java is one of the most used programming language by the developers. It is widely used because it is concurrent, class-based and object-oriented. “Write once, run anywhere” (WORA) meaning that, it can be run in any device that supports java without being recompiled. Java programs can be run on any Java Virtual Machine (JVM). Furthermore, Java is very much the first choice for any developers if they intend to make web applications based on client-server. Java was originally developed by James Gosling at Sun Microsystems on 1995 which is now acquired by Oracle Corporation. Much of Java’s syntax are derived from the other programming languages like C and C++ but its low-level facilities are fewer.

Fig 2.1. Logo of Java by Oracle Corporation.

2.1.2 Java Development Kit

Java Development Kit (JDK) is an implementation of Java SE or Java EE or Java ME platforms. It was also released by the Oracle Corporation. It is in the form of binary product. The target users of JDK are the Java developers on Linux, Mac OS, Windows or Solaris. JDK uses JVM and some other resources to finalize the development of a Java application. Today, JDK is the most used Software Development Kit (SDK). JDK itself has a collection of programming tools. Some of them are appletviewer, javac, javadoc, jar, JConsole, keytool, pack200, VisualVM. There are some other JDKs that are used in some other platforms. For example, Azul Systems for Linux, OpenJDK based on Zulu for Linux, Windows, Mac OS X, Oracle Corporation's JRockit JDK for Windows, Linux and Solaris, IBM J9 JDK for AIX, Linux, Windows, MVS, OS/400, Pocket PC, z/OS.

Fig. 2.2. Logo of JDK by Oracle Corporation.

2.1.3 Smith-Waterman algorithm

Smith-Waterman algorithm is a dynamic programming algorithm. It was developed in 1981 by Temple F. Smith and Machael Waterman. This algorithm performs Local alignment and used in the bioinformatics to align DNA or protein sequences to find out the similar regions of the given two sequences. Instead of aligning the entire length of two protein sequences, this algorithm finds the region of highest similarity between two protein sequences. This is potentially more biologically relevant due to the fact that the ends of protein sequences tend to be less highly conserved than the middle portions, leading to higher mutation, deletion, and insertion rates at the end of the protein sequence. The Smith-Waterman algorithm allows us to align proteins more accurately without having to align the ends of related protein which may be highly different [8]. The algorithm of the Smith-Waterman written in pseudo code is given below.

```
Initialization:
    $F (0, j) = 0$
    $F (i, 0) = 0$
Filling Matrix:
    for each i, j = 0 to M, N {
    $F (i, j) = max (0, F(i - 1,j - 1) + s, F(I - 1, j) – d, F(i, j - 1) – d)$ }
Traceback:
    $F_{opt} = max (F (i, j))$
    traceback ($F_{opt}$)
```

Here,

- M and N is the length of the two sequences

- F is the Matrix

- F (i, j) is the maximum Similarity-Score between a suffix of a[1...i] and a suffix of b[1...j]

- S and d are gap scoring

In the smith-Waterman algorithm to fill the matrix, the matrix for negative values is considered as 0 as potentially being the maximum value of the three other cases (where xi=yj, or there is a gap in x or a gap in y). By not letting any of the values go below zero, the algorithm stop considering regions of high dissimilarity which have no good alignments. This allows the algorithm to focus on only those regions of the protein which are similar. For traceback, this algorithm doesn't start at the n-terminus of both sequences, rather it starts at the cell with the highest score in the entire matrix. This allows the alignment of the similar subsequences of the proteins [8].

2.1.4 BLAST algorithm

BLAST, stands for Basic Local Alignment Search Tool, is an algorithm which compares the primary biological sequences information such as nucleotide of DNA sequences and different types of protein sequences. The BLAST algorithm and program were designed by Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers, and David J. Lipman at the National Institutes of Health and was published in the "Journal of Molecular Biology" in 1990 and cited over 50,000 times. A BLAST search enables a researcher to compare a query sequence with a library or database of sequences. This algorithm was developed to perform sequences similarity search of DNA or protein sequences that is faster than FASTA but equally sensitive [14]. BLAST use heuristic approach to find out the similarities which helps the program to run fast. It also provides an approximate output when the exact output doesn't exist.

For rapid reformatting of search results and enhancement of server, new queuing system has been implemented for BLAST which is called “QBLAST”. The BLAST algorithm has not changed; QBLAST simply offers a modular approach that separates the search step from the output formatting step [15]. An overview of the BLAST algorithm (a protein to protein search) is as follows.

1. Remove low-complexity region or sequence repeats in the query sequence. "Low-complexity region" means a region of a sequence composed of few kinds of elements. These regions might give high scores that confuse the program to find the actual significant sequences in the database, so they should be filtered out. The regions will be marked with an X (protein sequences) then would be ignored by the BLAST program. To filter out the low-complexity regions, the SEG program is used for protein sequences.
2. Make a k-letter word list of the query sequence.
3. List the possible matching words. BLAST only cares about the high-scoring words. The scores are created by comparing the word in the list in step 2 with all the 3-letter words. By using the scoring matrix (substitution matrix) to score the comparison of each residue pair, there are 20^3 possible match scores for a 3-letter word.
4. Organize the remaining high-scoring words into an efficient search tree. This allows the program to rapidly compare the high-scoring words to the database sequences.
5. Repeat step 3 to 4 for each k-letter word in the query sequence.
6. Scan the database sequences for exact matches with the remaining high-scoring words. The BLAST program scans the database sequences for the remaining high-scoring word, such as PEG, of each position. If an exact match is found, this match is used to seed a possible un-gapped alignment between the query and database sequences.
7. Extend the exact matches to high-scoring segment pair (HSP).
   a. The original version of BLAST stretches a longer alignment between the query and the database sequences in the left and right directions, from the

position where the exact match occurred. The extension does not stop until the accumulated total score of the HSP begins to decrease.

   b. To save more time, a newer version of BLAST, called BLAST2 or gapped BLAST, has been developed. BLAST2 adopts a lower neighborhood word score threshold to maintain the same level of sensitivity for detecting sequence similarity. Therefore, the possible matching words list in step 3 becomes longer. Next, the exact matched regions, within distance A from each other on the same diagonal in figure 3, will be joined as a longer new region. Finally, the new regions are then extended by the same method as in the original version of BLAST, and the HSPs' (High-scoring segment pair) scores of the extended regions are then created by using a substitution matrix as before.

8. List all of the HSPs in the database whose score is high enough to be considered. We list the HSPs whose scores are greater than the empirically determined cutoff score S. By examining the distribution of the alignment scores modeled by comparing random sequences, a cutoff score S can be determined such that its value is large enough to guarantee the significance of the remaining HSPs.
9. Evaluate the significance of the HSP score.
10. Make two or more HSP regions into a longer alignment. Sometimes, we find two or more HSP regions in one database sequence that can be made into a longer alignment. This provides additional evidence of the relation between the query and database sequence. There are two methods, the Poisson method and the sum-of-scores method, to compare the significance of the newly combined HSP regions.
11. Show the gapped Smith-Waterman local alignments of the query and each of the matched database sequences.
    a. The original BLAST only generates un-gapped alignments including the initially found HSPs individually, even when there is more than one HSP found in one database sequence.

b. BLAST2 produces a single alignment with gaps that can include all of the initially-found HSP regions. Note that the computation of the score and its corresponding *E*-value involves use of adequate gap penalties.

12. Report every match whose expect score is lower than a threshold parameter E.

2.2 Tools

2.2.1 PHP

PHP is a server-side scripting language designed for the developers for web development. It is a broadly used language by the developers to build different kind of websites. PHP stands for Personal Home Page. It now stands for the recursive backronym. PHP codes are basically embedded into HTML code. It can also be used by combining with various web template systems as well as different web frameworks. PHP codes are processed by a PHP interpreter that is a module in the web server.

Fig. 2.3. Logo of PHP.

2.2.2 HTML5

Hypertext Markup Language (HTML) is the standard language to create web pages. Web browsers can read HTML files and transfer them to visual or audible online web pages. HTML elements are like the building blocks for any website. By combining both PHP and HTML, a website is designed successfully with color being added by CSS. HTML is based on many attributes. It has hundreds of attributes to work with and more attributes are being generated over a short period of time. HTML5 is published on 28 October 2014 by the World Wide Web Consortium (W3C). It is the fifth version of the HTML standard. Its main focus is to support the latest multimedia and also to keep it simple for the user on World Wide Web. Some of the new features that HTML5 has included are <video>, <audio> and <canvas> elements in the attributes. Integration of scalable vector graphs (SVG) content and MathML are some of the major inclusion for mathematical formuli. These features are included so that, developers don’t need to have resort to property plugins and APIs.

Fig. 2.4. HTML 5 logo.

2.2.3 Eclipse

Eclipse was first launched in the November, 2001. It was then launched by IBM including seven other companies. Since then it is an open source project. Since it had been published, Eclipse has gone beyond every ounce of expectations. Still now it is one of the greatest developing platforms for the developers. Some of the major projects are now run by eclipse.org. Hundreds of thousands of products are built on eclipse platform till date. Lots of institutions around the world are now using Eclipse to teach students Java. With all these great assets eclipse.org is still an open source learning material for all the developers out there in the current world [7]. Eclipse updates their versions quite regularly. Though there are newer versions of Eclipse, Eclipse Juno was used throughout the coding period with no much great importance given on the selection of versions to be used.

2.2.4 Eclipse ADT (Android Development Tools)

To build Android applications, Google has provided a plugin for the Eclipse IDE that is called the Eclipse ADT (Android Development Tools). It gives the developer access in the deigning and coding part of the Android world of applications.

Fig. 2.5. Eclipse logo.

2.2.5 XAMPP

XAMPP is a widely used server which is free and runs on open source cross-platforms. It was developed by the Apache Friends. It consists the Apache HTTP Server, MariaDB database and scripts for PHP and Perl Programming Language. XAMPP basically stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P), Perl (P). XAMPP is widely used by the web developers to create temporary servers to test their websites. XAMPP uses a localhost with the IP address 127.0.0.1. XAMPP derives the personal computer to a local server and can be reached under the same network to test the websites as a temporary domain.

Fig. 2.6. XAMPP logo.

CHAPTER III: Results

3. RESULTS

3.1 Comparison between programming languages

As from reference no. [6], a survey result shows comparisons among some of the languages that are usually used to code in bioinformatics. These languages are C, C++, C#, Java, Perl and Python. Perl and Python were the slowest amongst those six languages that were compared for local alignment program. For the same code and program that were run on Linux and Windows respectively gave the fastest outcome for C, C++, C# and Java written code.

For BLAST parsing program C was the fastest. The other three languages C++, Java and Perl are relatively slow and took almost the same amount of time. Among the six languages C# and Python were the slowest. All the languages were faster on Linux platform in comparison to Windows platform.

Perl and Python consumed the most memory in terms of local alignment. C, C++ C# and Java consumed the same amount of memory but all of them were pretty much memory efficient in comparison to Perl and Python.

For BLAST, C and C++ consumed the highest amount of memory. On the other hand, C#, Java, Perl and Python were less consuming and effectively more efficient. All of them were run on both Linux and Windows platform but didn't differ much.

Results of these surveys effectively shown that, C was the best performer both in terms of speed and memory usage but that effectively took more lines of codes to be written as it doesn't use much standard libraries. On the other hand, Java is a portable web oriented language. Sun is consistently improving the Java compiler and interpreter and other JVM implementations. Moreover, Java uses many standard libraries which affects the performance and memory usage greatly and also reduces the huge amount of lines of codes. Java also provides with the advantage of running the program on any machine that has JDK. As a result, Java was chosen to write the raw level code of this project from the scratch.

3.2 Programming of different algorithms

After the selection of the Java language over the other languages e.g. C, C++, C#, Perl and Python, raw Java code was written for all the three algorithms, Smith-Waterman, Needleman-Wunsch and BLAST. Some of the Java APIs that were used are given below.

```
org.biojava.nbio.ws.alignment.qblast.BlastAlignmentParameterEnum.ENTREZ_QUERY;

org.biojava.nbio.core.sequence.io.util.IOUtils;

org.biojava.nbio.ws.alignment.qblast.*;

java.nio.file.Files;

java.nio.file.Paths;
```

These codes were written so that a comparison can be made among the three algorithms and the best suited one can be selected for the mobile application development.

3.3 Comparison between algorithms

The challenge in performing sequence alignments has been the tradeoff between accuracy and efficiency [8]. There are many different algorithms available for sequence alignments. Among them most commonly used algorithms are BLAST, Smith-Waterman, Needleman-Wunsch algorithm etc. Finding optimal alignment and high computational complexity is the main feature of Needleman-Wunsch and Smith-Waterman algorithms. On the other hand, BLAST is much faster than the other two in giving response to any query. While choosing among these algorithms, the faster responding ones were preferred which were able to handle the ever increasing vast database of DNA and protein sequences [8].

Needleman-Wunsch and Smith-Waterman algorithm are almost same. They have the same space complexity. In addition, both of these algorithms are dynamic, which ensures a similar reaction time to any query. Smith-Waterman and Needleman-Wunsch algorithms are very much time consuming and also require strong support from computer power. BLAST is a very well regarded algorithm to produce results rapidly [9]. BLAST also uses a heuristic approach which is significantly faster than the dynamic programming algorithms [8]. The time complexity, space complexity and programming method of BLAST, Smith-Waterman and Needleman-Wunsch are given below in the table (3.1).

Table 3.1

Comparisons among Smith-Waterman, Needleman-Wunsch and BLAST algorithms.

Name of the algorithm	Time complexity	Space complexity	Programming	Computation complexity
BLAST	O(MN)	O(20w + MN)	Heuristic	O(20w)
Smith-Waterman	O(MN)	O(MN)	Dynamic	O(MN)
Needleman-Wunsch	O(MN)	O(MN)	Dynamic	O(MN)

Table 3.1 shows the time complexity of BLAST, Smith-Waterman and Needleman-Wunsch. They are quite the same and have identical time complexity and space complexity, O(MN). For space complexity, BLAST has the highest value over Smith-Waterman and Needleman-Wunsch. But all of them require the same amount of space to run the program. Dynamic algorithm is a recursive process and by avoiding multiple calculations it should be able to save the value that already has been calculated thus the algorithm will run faster than ever [10]. Heuristic programming is good for feasible and satisfactory solutions in a few seconds [11]. Lastly, BLAST provides low computational complexity O$(20^w + MN)$ compared to other two algorithms which require O(MN) .

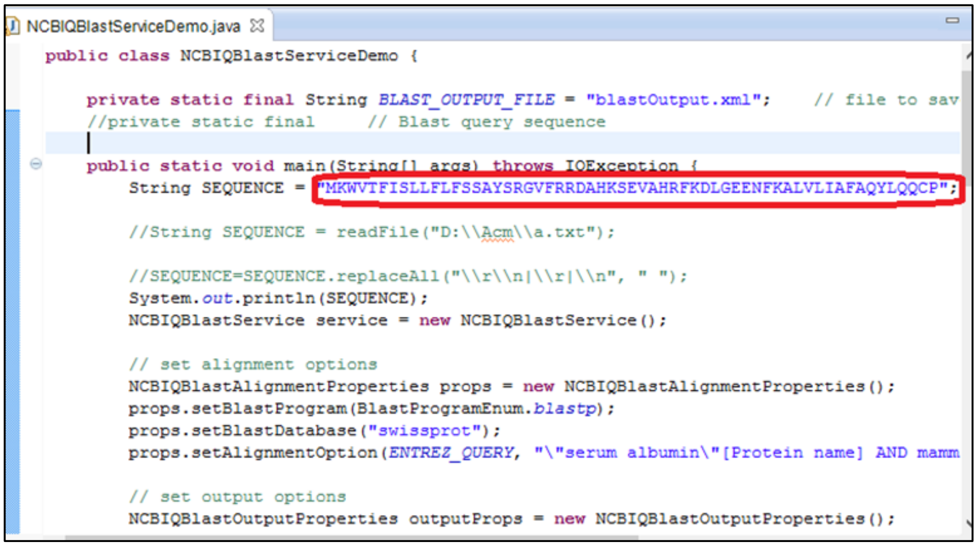
In addition to the above values, all three algorithms were put into test and scores were evaluated. "The alignment with the largest score must be the optimal alignment."[8]. As a result, by inserting the same sequence, some results were tested out and observed, which one produces the largest score in the system.

The input of protein sequences are given below:

Sequence 1:
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP

Sequence 2:
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP

With the help of the NCBI database server, we had run the BLAST algorithm to find out the matched sequence. Input sequence is in the red circle (Figure 3.1).

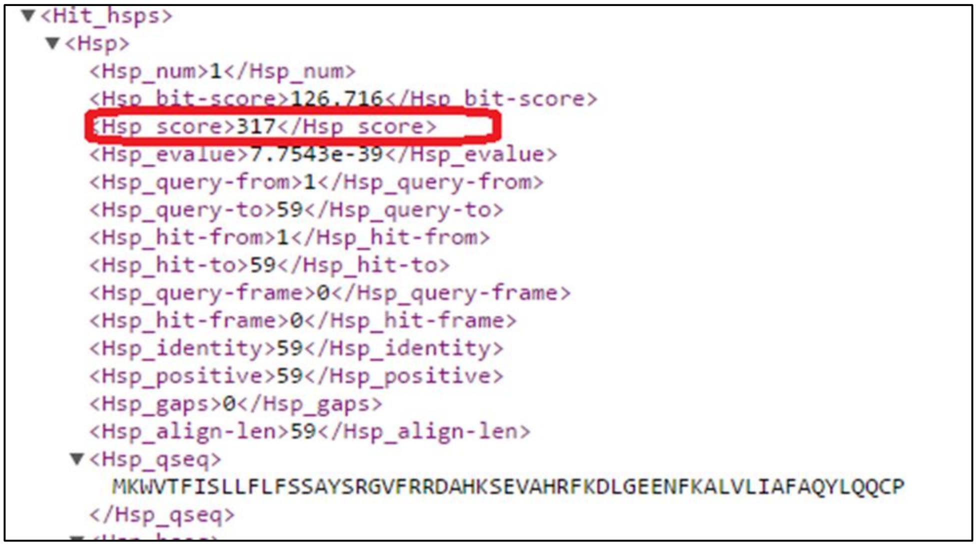


NCBIQBlastServiceDemo.java

```java
public class NCBIQBlastServiceDemo {

    private static final String BLAST_OUTPUT_FILE = "blastOutput.xml";    // file to sav
    //private static final     // Blast query sequence

    public static void main(String[] args) throws IOException {
        String SEQUENCE = "MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP";

        //String SEQUENCE = readFile("D:\\Acm\\a.txt");

        //SEQUENCE=SEQUENCE.replaceAll("\\r\\n|\\r|\\n", " ");
        System.out.println(SEQUENCE);
        NCBIQBlastService service = new NCBIQBlastService();

        // set alignment options
        NCBIQBlastAlignmentProperties props = new NCBIQBlastAlignmentProperties();
        props.setBlastProgram(BlastProgramEnum.blastp);
        props.setBlastDatabase("swissprot");
        props.setAlignmentOption(ENTREZ_QUERY, "\"serum albumin\"[Protein name] AND mamm

        // set output options
        NCBIQBlastOutputProperties outputProps = new NCBIQBlastOutputProperties();
```

Fig. 3.1. Running BLAST algorithm by giving a sequence.

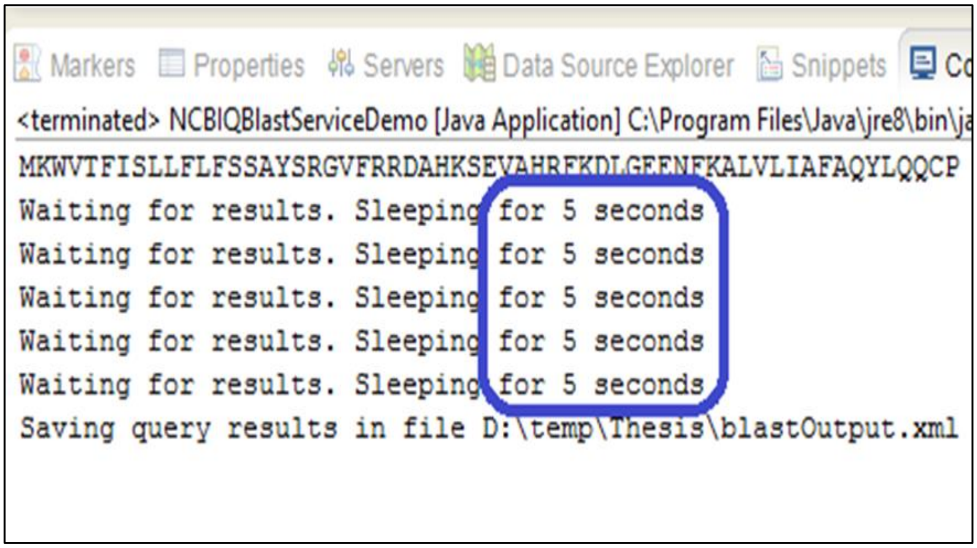
The highest score was 317 (mark in red circle) with the same sequence string as shown in (Figure 3.2).



```
▼<Hit_hsps>
  ▼<Hsp>
     <Hsp_num>1</Hsp_num>
     <Hsp_bit-score>126.716</Hsp_bit-score>
     <Hsp_score>317</Hsp_score>
     <Hsp_evalue>7.7543e-39</Hsp_evalue>
     <Hsp_query-from>1</Hsp_query-from>
     <Hsp_query-to>59</Hsp_query-to>
     <Hsp_hit-from>1</Hsp_hit-from>
     <Hsp_hit-to>59</Hsp_hit-to>
     <Hsp_query-frame>0</Hsp_query-frame>
     <Hsp_hit-frame>0</Hsp_hit-frame>
     <Hsp_identity>59</Hsp_identity>
     <Hsp_positive>59</Hsp_positive>
     <Hsp_gaps>0</Hsp_gaps>
     <Hsp_align-len>59</Hsp_align-len>
   ▼<Hsp_qseq>
       MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
     </Hsp_qseq>
```

Fig. 3.2. Results after running BLAST algorithm, in XML format.

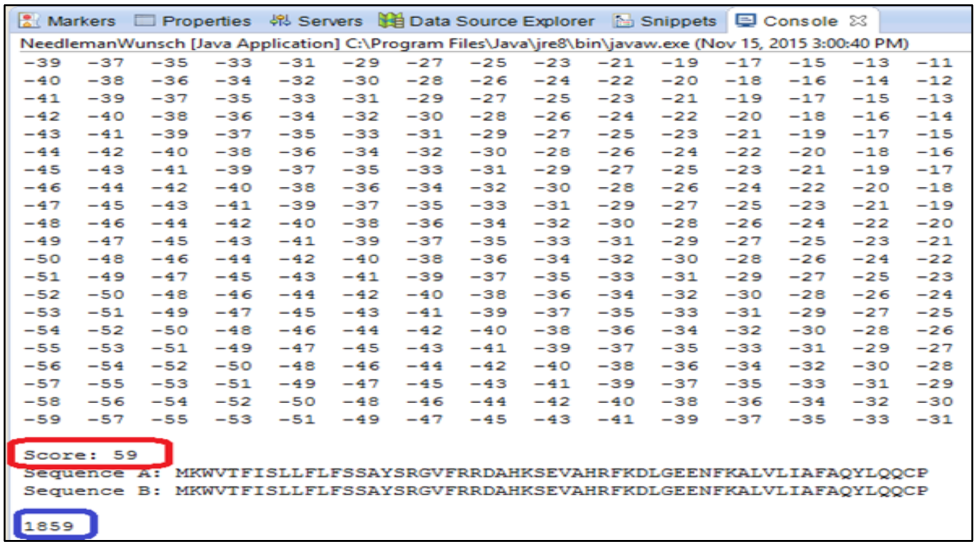
To run the entire database, for this particular sequence BTAST took 25 seconds (Figure 3.3).



```
<terminated> NCBIQBlastServiceDemo [Java Application] C:\Program Files\Java\jre8\bin\ja
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
Waiting for results. Sleeping for 5 seconds
Waiting for results. Sleeping for 5 seconds
Waiting for results. Sleeping for 5 seconds
Waiting for results. Sleeping for 5 seconds
Waiting for results. Sleeping for 5 seconds
Saving query results in file D:\temp\Thesis\blastOutput.xml
```

Fig. 3.3: BLAST analysis with time record.

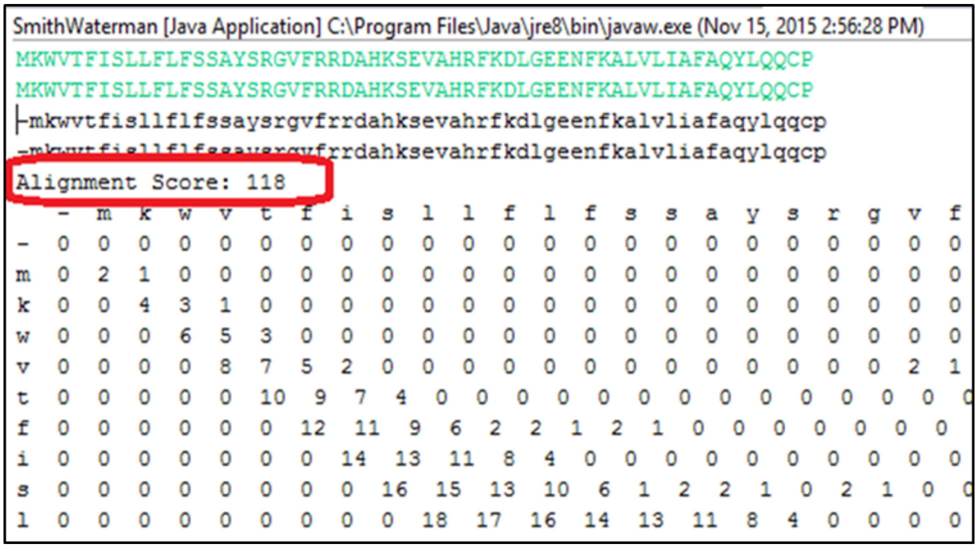
The same sequence was inserted in the Needlmen-Wunsch algorithm and the score was 59 (marked in red circle) and to compare with only one string, it took about 2 seconds (marked in blue circle). Detailed result is given below (Figure 3.4).



```
NeedlemanWunsch [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Nov 15, 2015 3:00:40 PM)
-39  -37  -35  -33  -31  -29  -27  -25  -23  -21  -19  -17  -15  -13  -11
-40  -38  -36  -34  -32  -30  -28  -26  -24  -22  -20  -18  -16  -14  -12
-41  -39  -37  -35  -33  -31  -29  -27  -25  -23  -21  -19  -17  -15  -13
-42  -40  -38  -36  -34  -32  -30  -28  -26  -24  -22  -20  -18  -16  -14
-43  -41  -39  -37  -35  -33  -31  -29  -27  -25  -23  -21  -19  -17  -15
-44  -42  -40  -38  -36  -34  -32  -30  -28  -26  -24  -22  -20  -18  -16
-45  -43  -41  -39  -37  -35  -33  -31  -29  -27  -25  -23  -21  -19  -17
-46  -44  -42  -40  -38  -36  -34  -32  -30  -28  -26  -24  -22  -20  -18
-47  -45  -43  -41  -39  -37  -35  -33  -31  -29  -27  -25  -23  -21  -19
-48  -46  -44  -42  -40  -38  -36  -34  -32  -30  -28  -26  -24  -22  -20
-49  -47  -45  -43  -41  -39  -37  -35  -33  -31  -29  -27  -25  -23  -21
-50  -48  -46  -44  -42  -40  -38  -36  -34  -32  -30  -28  -26  -24  -22
-51  -49  -47  -45  -43  -41  -39  -37  -35  -33  -31  -29  -27  -25  -23
-52  -50  -48  -46  -44  -42  -40  -38  -36  -34  -32  -30  -28  -26  -24
-53  -51  -49  -47  -45  -43  -41  -39  -37  -35  -33  -31  -29  -27  -25
-54  -52  -50  -48  -46  -44  -42  -40  -38  -36  -34  -32  -30  -28  -26
-55  -53  -51  -49  -47  -45  -43  -41  -39  -37  -35  -33  -31  -29  -27
-56  -54  -52  -50  -48  -46  -44  -42  -40  -38  -36  -34  -32  -30  -28
-57  -55  -53  -51  -49  -47  -45  -43  -41  -39  -37  -35  -33  -31  -29
-58  -56  -54  -52  -50  -48  -46  -44  -42  -40  -38  -36  -34  -32  -30
-59  -57  -55  -53  -51  -49  -47  -45  -43  -41  -39  -37  -35  -33  -31

Score:  59
Sequence A: MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
Sequence B: MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP

1859
```

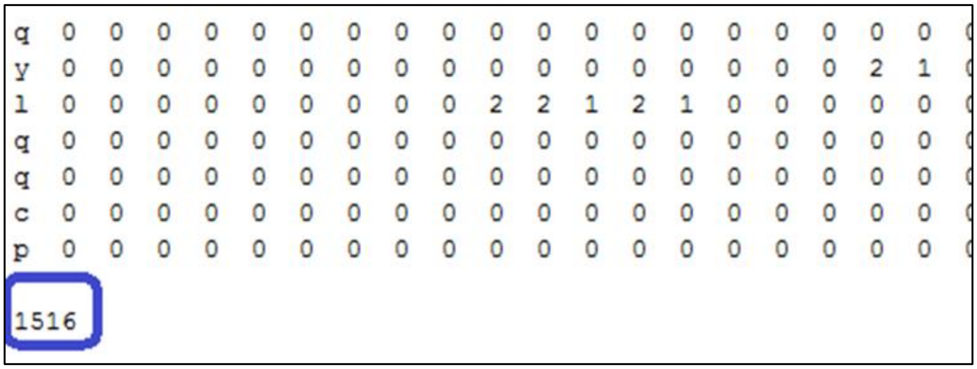
Fig. 3.4. Result of the Needleman-Wunsch algorithm with time recorded.

Finally, the same strings were tested on the Smith-Waterman algorithm and the score was 118 (marked in red circle) (Figure 3.5). Apart from that, it took 1 second to compare two sequences (marked in blue circle) (Figure 3.6).



```
SmithWaterman [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Nov 15, 2015 2:56:28 PM)
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
-mkwvtfisllflfssaysrgvfrrdahksevahrfkdlgeenfkalvliafaqylqqcp
-mkwvtfisllflfssaysrgvfrrdahksevahrfkdlgeenfkalvliafaqylqqcp
Alignment Score: 118
     -   m   k   w   v   t   f   i   s   l   l   f   l   f   s   s   a   y   s   r   g   v   f
-    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
m    0   2   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
k    0   0   4   3   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
w    0   0   0   6   5   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
v    0   0   0   0   8   7   5   2   0   0   0   0   0   0   0   0   0   0   0   0   0   2   1
t    0   0   0   0   0  10   9   7   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0
f    0   0   0   0   0   0  12  11   9   6   2   2   1   2   1   0   0   0   0   0   0   0
i    0   0   0   0   0   0   0  14  13  11   8   4   0   0   0   0   0   0   0   0   0   0   0
s    0   0   0   0   0   0   0   0  16  15  13  10   6   1   2   2   1   0   2   1   0   0
l    0   0   0   0   0   0   0   0   0  18  17  16  14  13  11   8   4   0   0   0   0
```

Fig. 3.5. Result of Smith-Waterman algorithm.



```
q    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
y    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   1
l    0   0   0   0   0   0   0   0   0   2   2   1   2   1   0   0   0   0   0
q    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
q    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
c    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
p    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

1516
```

Fig. 3.6. Result of Smith-Waterman algorithm with time recorded.

The above results demonstrate that, BLAST was the most efficient and appropriate algorithm as it had given the highest score. Moreover, running a sequence alignment search against the whole database took the least amount of time in BLAST. On the other hand, Smith-Waterman and Needleman-Wunsch took almost 2 seconds for only one string comparison. Along with that Smith-Waterman and Needleman algorithms were also applicable for local alignment while BLAST could be applied for both local and multiple alignments. For these reasons BLAST algorithm was considered for future use in the smart phone application system.

3.4 Selection of the main features for the application

Feature selection is a process in which subsets of available features are selected for application in prediction models [12]. Two major features of the mobile application were (i) analyzing local alignment and (ii) BLAST analysis. By giving protein sequences or accession IDs, users could analyze the BLAST algorithm and get the result from the NCBI website accurately. Providing two either DNA sequences or protein sequences, users can analyze local alignment and get the matched score if positive or zero.

3.5 Programming for the mobile application and features adjustment

The basic programming was done on the Eclipse Juno ADT. The whole coding part was started from the scratch. The application had six layouts including a splash screen that glorifies the application name “BLAST Sequence”. After the splash screen, the main activity layout would show two main features of the application, “Local Alignment” and “BLAST”.

Smith-Waterman algorithm was used to write the code for the local alignment. Clicking on the “Local Alignment” button would prompt the user to the local alignment activity where the user would be able to enter two sequences either DNA or protein of his/her like. After the successful inclusion of the two sequences the user may proceed to click the “Get Score” button which will show the result on the bottom of the screen “Alignment Score: xxx”.

The “BLAST” button would prompt the user to the BLAST activity where it would ask the user to insert a sequence of any length on the search bar. Then the user might proceed to the “BLAST” button. The “BLAST” button sent the sequence to the NCBI server. Then the server responded after a few moments. After some time, there would came a toast that would say where the result would be saved on the server as well as the time that the app was on sleep in the process of getting the result from the NCBI website. Then there would appear a “Result” button. After clicking the “Result” button the user would be prompted to a new activity layout where a “Parse” button would appear. This “Parse” button usually parses the result from the NCBI website. Clicking of the “Parse” button the user would get the result where he would find the best matched sequences in a decreasing order. The following information would be provided in this layout:

a. Query information (Accession ID, Organism name)
b. Identity
c. E-value
d. Bit score
e. Matched query sequence
f. Score

All the features of the mobile application are given below (Figure 3.7).

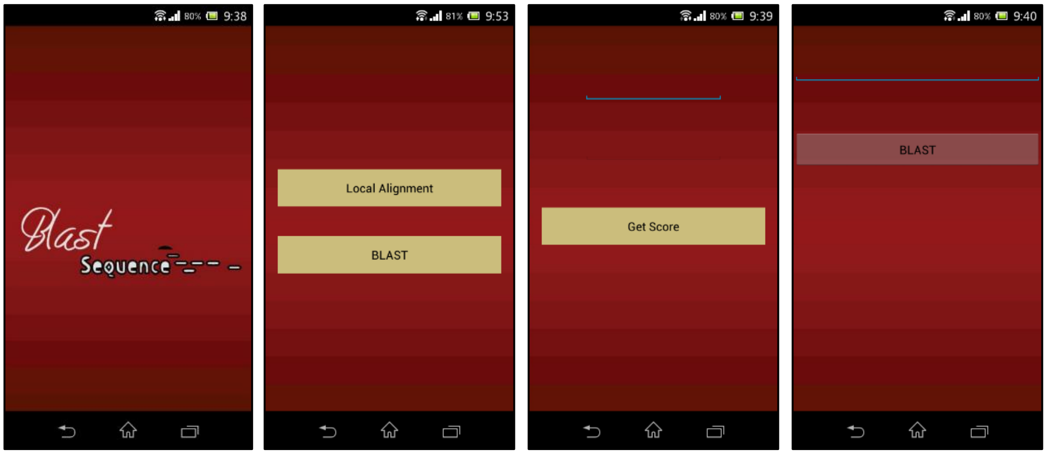


Fig. 3.7. Mobile application and its basic features.

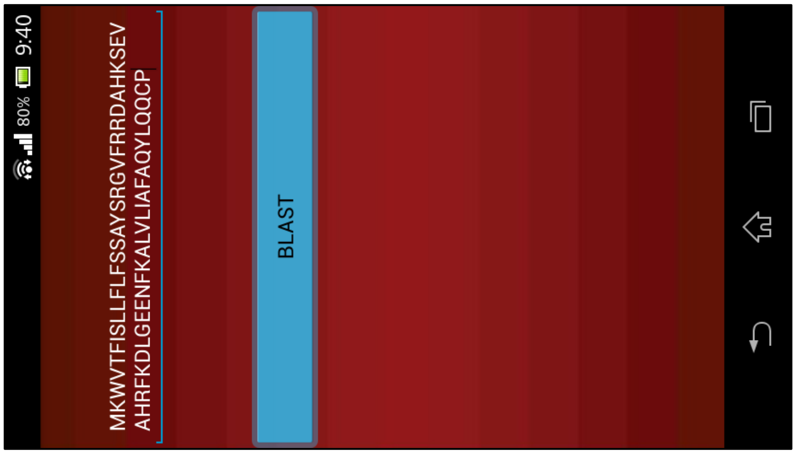
3.6 Programming for the server

The programming part of the server was done in PHP and HTML5. XAMPP was used to build a local server. The server part basically would get the input sequence from the user from the mobile application “BLAST” activity then it would send it to the NCBI website server. After getting the response and the result from the NCBI website, the results would be saved on the server database. When the user clicks on the “Result” button of the mobile application, the server then would parse the result from the database and then would sent the results to the “Result” layout.

3.7 Evaluation of the application with computer system

Results were evaluated by comparing the results of the computer system and the mobile system. Same protein sequences were given in both the computer and the mobile application. Analysis of the sequences using BLAST is given below (Figure 3.8).

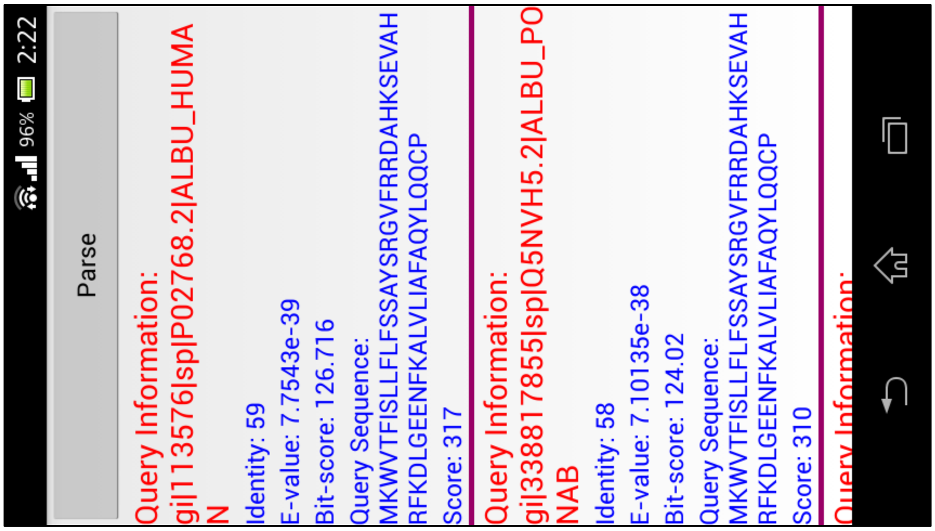
Input Sequence:
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYL
QQCP



MKWVTFISLLFLFSSAYSRGVFRRDAHKSEV
AHRFKDLGEENFKALVLIAFAQYLQQCP

Fig. 3.8. The DNA sequence is given as input and the BLAST button is clicked for analysis.

The results from the mobile application (Figure 3.9) and the computer system are given below (Figure 3.10).

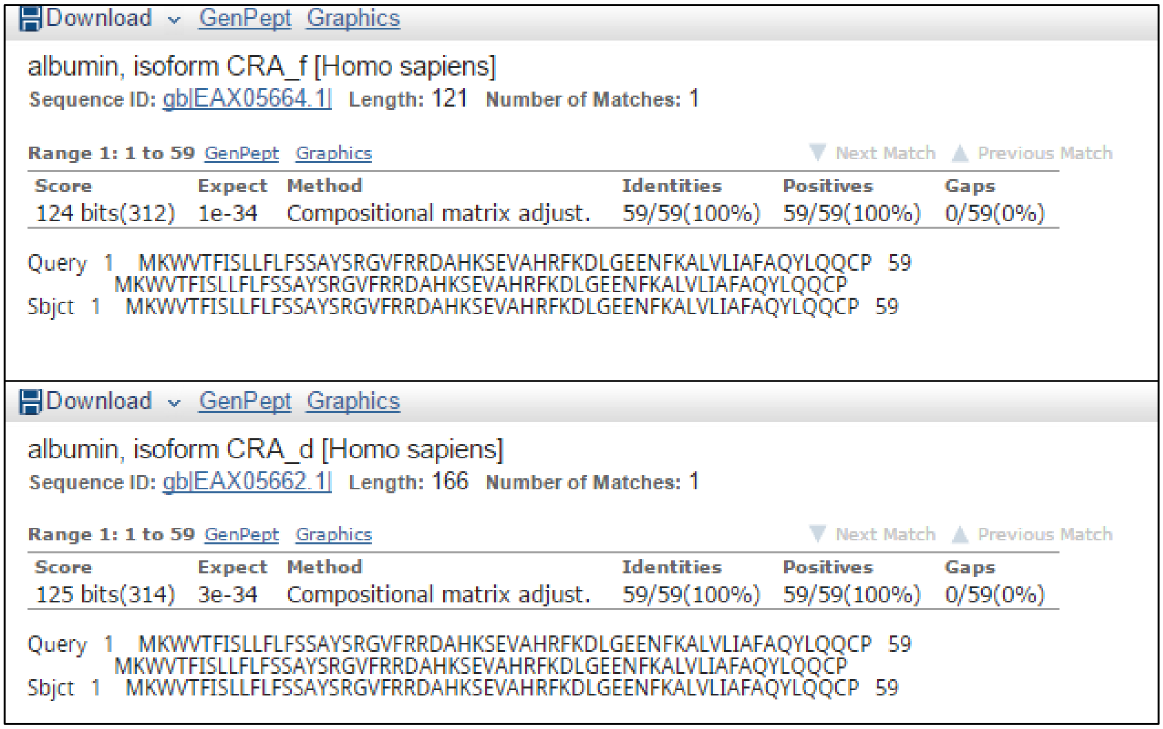


Query Information:
gi|113576|sp|P02768.2|ALBU_HUMAN
Identity: 59
E-value: 7.7543e-39
Bit-score: 126.716
Query Sequence:
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
Score: 317

Query Information:
gi|338817855|sp|Q5NVH5.2|ALBU_PONAB
Identity: 58
E-value: 7.10135e-38
Bit-score: 124.02
Query Sequence:
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
Score: 310

Query Information:

Fig. 3.9. Results of BLAST analysis in the mobile application.



albumin, isoform CRA_f [Homo sapiens]
Sequence ID: gb|EAX05664.1| Length: 121 Number of Matches: 1

Range 1: 1 to 59 GenPept Graphics

Score	Expect	Method	Identities	Positives	Gaps
124 bits(312)	1e-34	Compositional matrix adjust.	59/59(100%)	59/59(100%)	0/59(0%)

```
Query  1   MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP  59
           MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
Sbjct  1   MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP  59
```

albumin, isoform CRA_d [Homo sapiens]
Sequence ID: gb|EAX05662.1| Length: 166 Number of Matches: 1

Range 1: 1 to 59 GenPept Graphics

Score	Expect	Method	Identities	Positives	Gaps
125 bits(314)	3e-34	Compositional matrix adjust.	59/59(100%)	59/59(100%)	0/59(0%)

```
Query  1   MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP  59
           MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP
Sbjct  1   MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP  59
```

Fig. 3.10. BLAST results from computer system.

Results could also be driven from the mobile system using the Accession ID in the BLAST layout where previously the sequence was put (Figure 3.11).

Accession ID: EAX05664.1

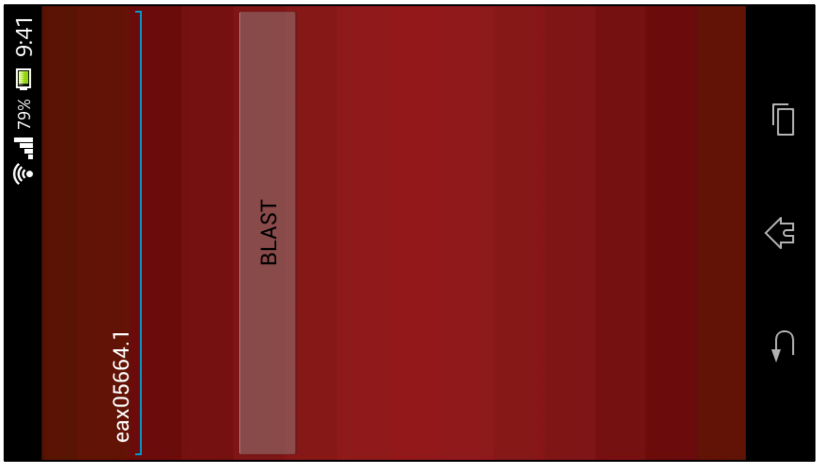
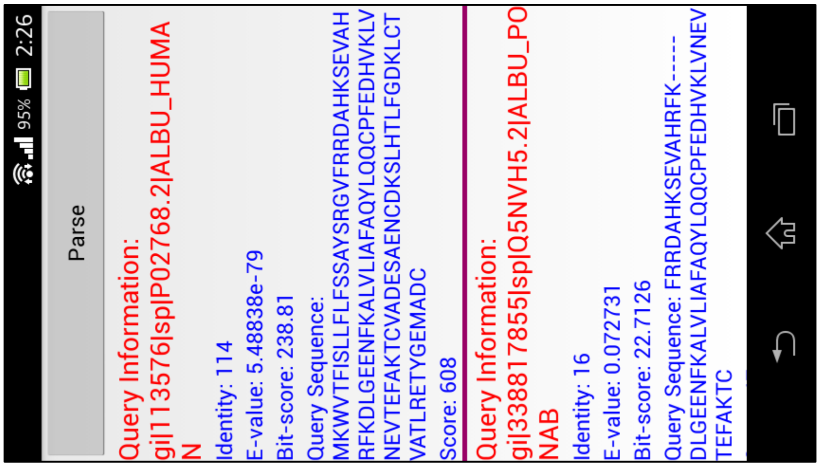


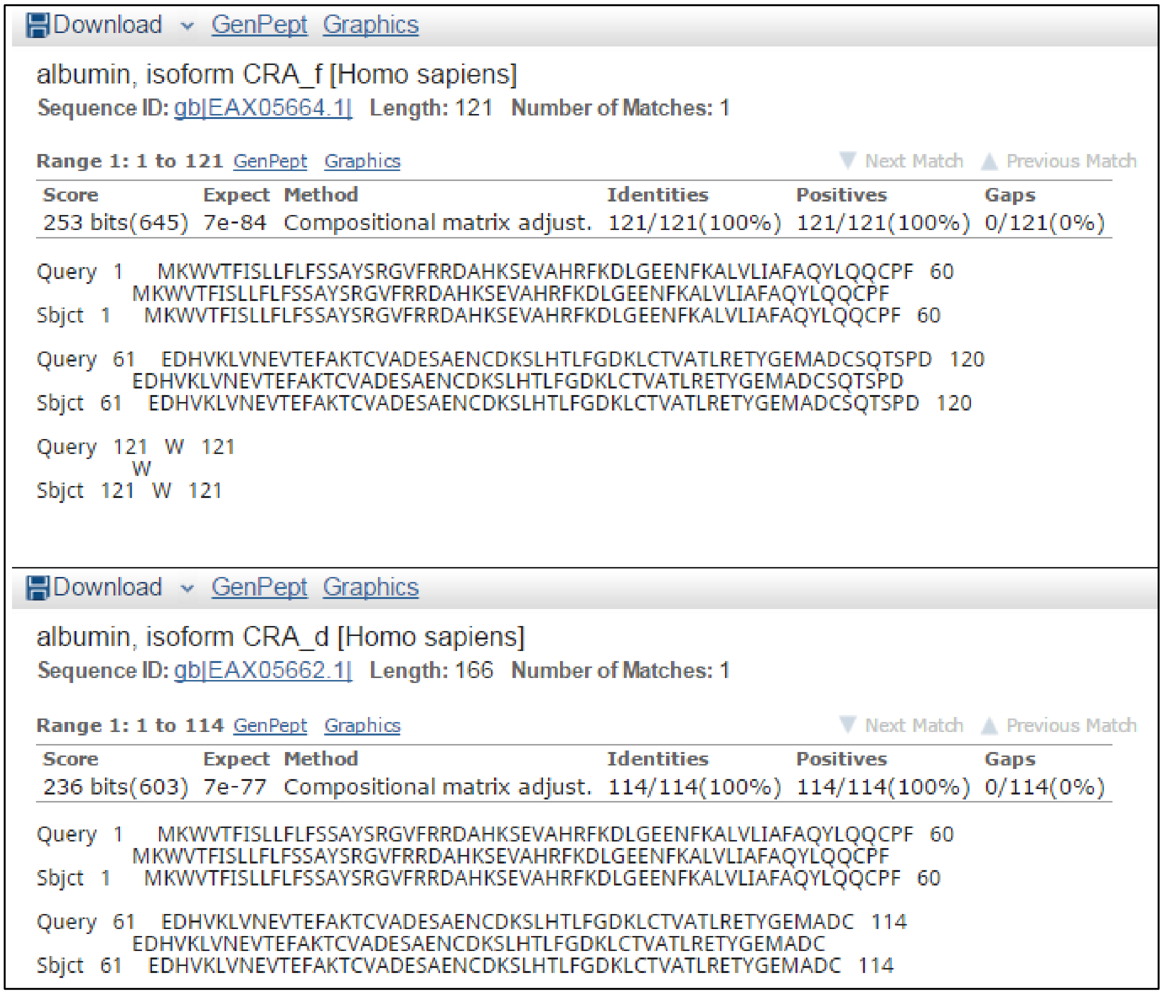
Fig. 3.11. The Accession ID is given as input and the BLAST button is clicked for analysis.

Results are shown below from android application (Figure 3.12) and computer systems (Figure 3.13).



```
Query Information:
gi|113576|sp|P02768.2|ALBU_HUMA
N
Identity: 114
E-value: 5.48838e-79
Bit-score: 238.81
Query Sequence:
MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAH
RFKDLGEENFKALVLIAFAQYLQQCPFEDHVKLV
NEVTEFAKTCVADESAENCDKSLHTLFGDKLCT
VATLRETYGEMADC
Score: 608
Query Information:
gi|338817855|sp|Q5NVH5.2|ALBU_PO
NAB
Identity: 16
E-value: 0.072731
Bit-score: 22.7126
Query Sequence: FRRDAHKSEVAHRFK-----
DLGEENFKALVLIAFAQYLQQCPFEDHVKLVNEV
TEFAKTC
```

Fig. 3.12. Results of BLAST analysis in the mobile application.



albumin, isoform CRA_f [Homo sapiens]

Sequence ID: gb|EAX05664.1| Length: 121 Number of Matches: 1

```
Range 1: 1 to 121 GenPept  Graphics                              ▼ Next Match  ▲ Previous Match
 Score          Expect  Method                         Identities      Positives       Gaps
 253 bits(645)  7e-84   Compositional matrix adjust.   121/121(100%)   121/121(100%)   0/121(0%)

Query  1    MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCPF  60
            MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCPF
Sbjct  1    MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCPF  60

Query  61   EDHVKLVNEVTEFAKTCVADESAENCDKSLHTLFGDKLCTVATLRETYGEMADCSQTSPD  120
            EDHVKLVNEVTEFAKTCVADESAENCDKSLHTLFGDKLCTVATLRETYGEMADCSQTSPD
Sbjct  61   EDHVKLVNEVTEFAKTCVADESAENCDKSLHTLFGDKLCTVATLRETYGEMADCSQTSPD  120

Query  121  W  121
            W
Sbjct  121  W  121
```

albumin, isoform CRA_d [Homo sapiens]

Sequence ID: gb|EAX05662.1| Length: 166 Number of Matches: 1

```
Range 1: 1 to 114 GenPept  Graphics                              ▼ Next Match  ▲ Previous Match
 Score          Expect  Method                         Identities      Positives       Gaps
 236 bits(603)  7e-77   Compositional matrix adjust.   114/114(100%)   114/114(100%)   0/114(0%)

Query  1    MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCPF  60
            MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCPF
Sbjct  1    MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCPF  60

Query  61   EDHVKLVNEVTEFAKTCVADESAENCDKSLHTLFGDKLCTVATLRETYGEMADC  114
            EDHVKLVNEVTEFAKTCVADESAENCDKSLHTLFGDKLCTVATLRETYGEMADC
Sbjct  61   EDHVKLVNEVTEFAKTCVADESAENCDKSLHTLFGDKLCTVATLRETYGEMADC  114
```

Fig. 3.13. BLAST results from computer system.

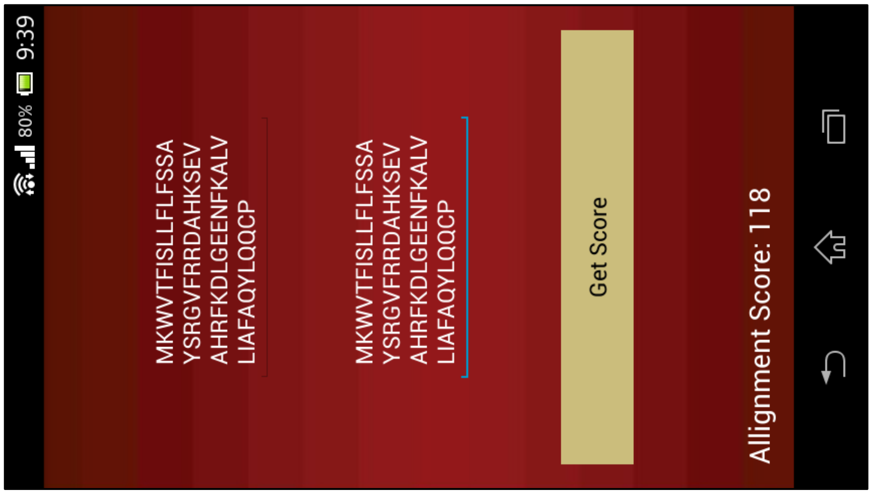
For the local alignment, two protein sequences were given and the score was checked if positive or zero (Figure 3.14).

Input Sequence 1:

MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP

Input Sequence 2:

MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP



MKWVTFISLLFLFSSA
YSRGVFRRDAHKSEV
AHRFKDLGEENFKALV
LIAFAQYLQQCP

MKWVTFISLLFLFSSA
YSRGVFRRDAHKSEV
AHRFKDLGEENFKALV
LIAFAQYLQQCP

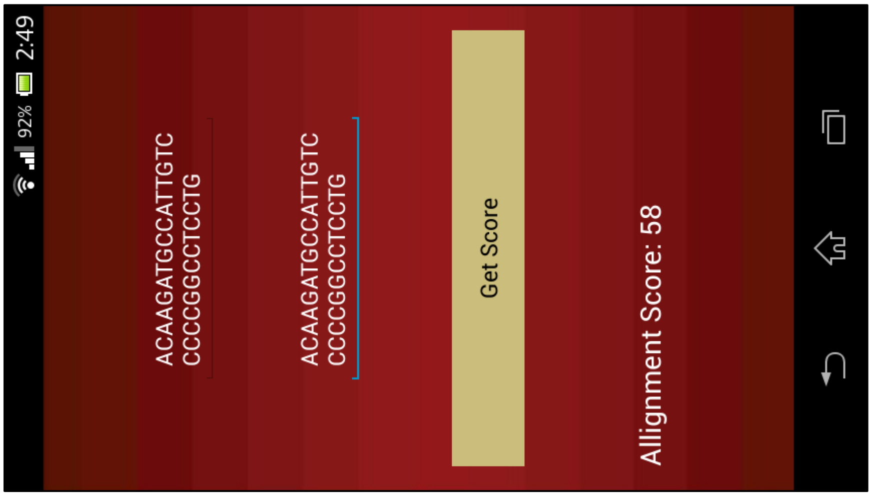
Allignment Score: 118

Fig. 3.14. Protein input sequences are given to generate score.

For the given protein sequences, the result score was 118. That means there are similarities between the given sequences. For the local alignment, two DNA sequences were given and the score was checked if positive or zero (Figure 3.15).

Input sequence 1: ACAAGATGCCATTGTCCCCCGGCCTCCTG

Input sequence 2: ACAAGATGCCATTGTCCCCCGGCCTCCTG



ACAAGATGCCATTGTC
CCCCGGCCTCCTG

ACAAGATGCCATTGTC
CCCCGGCCTCCTG

Allignment Score: 58

Fig. 3.15. DNA input sequences are given to generate score.

From the above image, it is clear that the sequence is matched to the input sequence as a result the output is greater than ‘0’. If there wasn’t any match between the inputs DNA sequences then there wouldn’t have been any score to show except ‘0’. The result can only be ‘0’ when there isn’t any match between the input sequences. For these particular sequences the result is 58 so the user can take these sequences for further research.

3.7.1 Evaluation of results for some more protein sequences and accession IDs:

Similarly some other inputs were given and the results were recorded. Images of the results of both the mobile application alongside the computer systems are given below to check for accuracy (Figure 3.16 – 3.21).

Sequences:

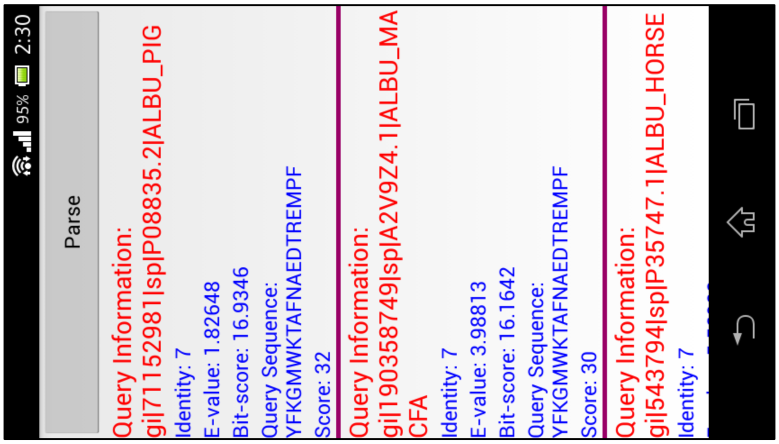
a. QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQES KPVQMMCMNNSFNVATLPAE
b. KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKR RVKVYLPQMKIEEKYNLTS
c. VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTG VIEDIKHSPESEQFRADHPFLFLIKHNPTNTIVYFGRYWSP

Accession IDs:

a. NP_001034065.1
b. XP_004429030.1
c. KQL88370.1

Sequence:

QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQ MMCMNNSFNVATLPAE



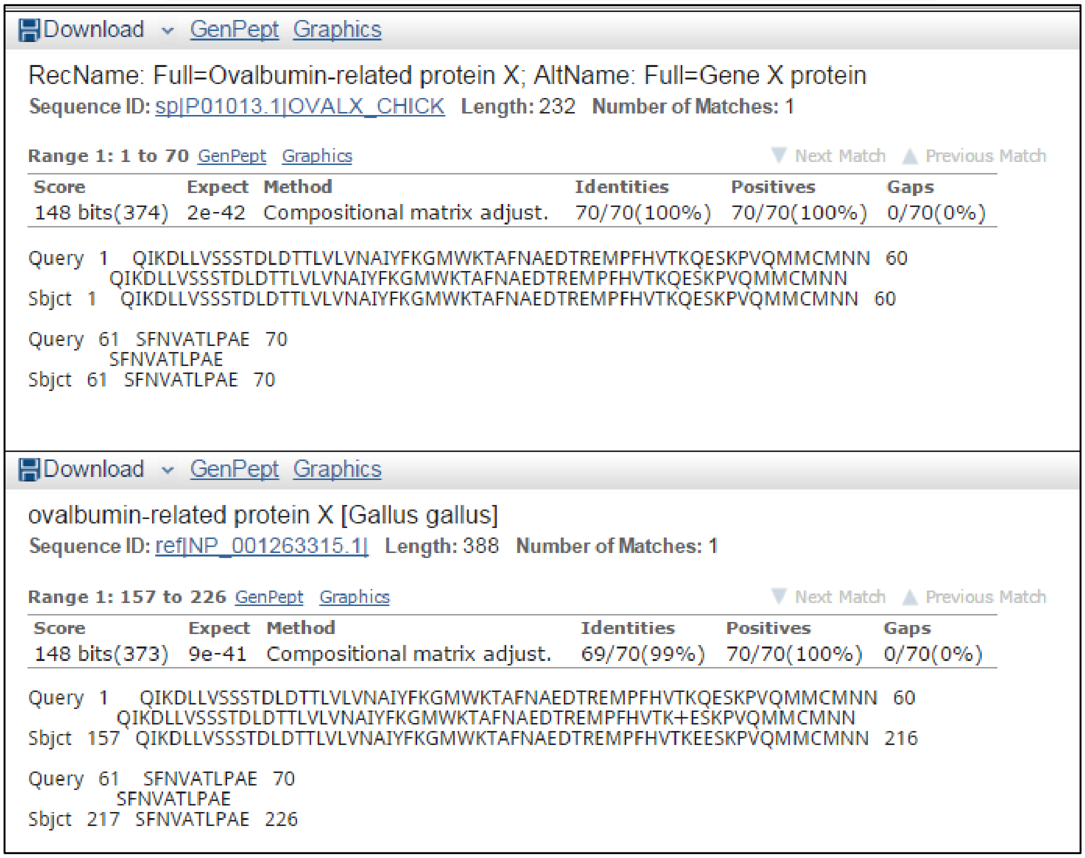
2:30 95%

Parse

Query Information:
gi|71152981|sp|P08835.2|ALBU_PIG
Identity: 7
E-value: 1.82648
Bit-score: 16.9346
Query Sequence:
YFKGMWKTAFNAEDTREMPF
Score: 32

Query Information:
gi|190358749|sp|A2V9Z4.1|ALBU_MA
CFA
Identity: 7
E-value: 3.98813
Bit-score: 16.1642
Query Sequence:
YFKGMWKTAFNAEDTREMPF
Score: 30

Query Information:
gi|543794|sp|P35747.1|ALBU_HORSE
Identity: 7



RecName: Full=Ovalbumin-related protein X; AltName: Full=Gene X protein
Sequence ID: sp|P01013.1|OVALX_CHICK Length: 232 Number of Matches: 1

Range 1: 1 to 70 GenPept Graphics

Score	Expect	Method	Identities	Positives	Gaps
148 bits(374)	2e-42	Compositional matrix adjust.	70/70(100%)	70/70(100%)	0/70(0%)

```
Query  1    QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMMCMNN  60
            QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMMCMNN
Sbjct  1    QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMMCMNN  60

Query  61   SFNVATLPAE  70
            SFNVATLPAE
Sbjct  61   SFNVATLPAE  70
```

ovalbumin-related protein X [Gallus gallus]
Sequence ID: ref|NP_001263315.1| Length: 388 Number of Matches: 1

Range 1: 157 to 226 GenPept Graphics

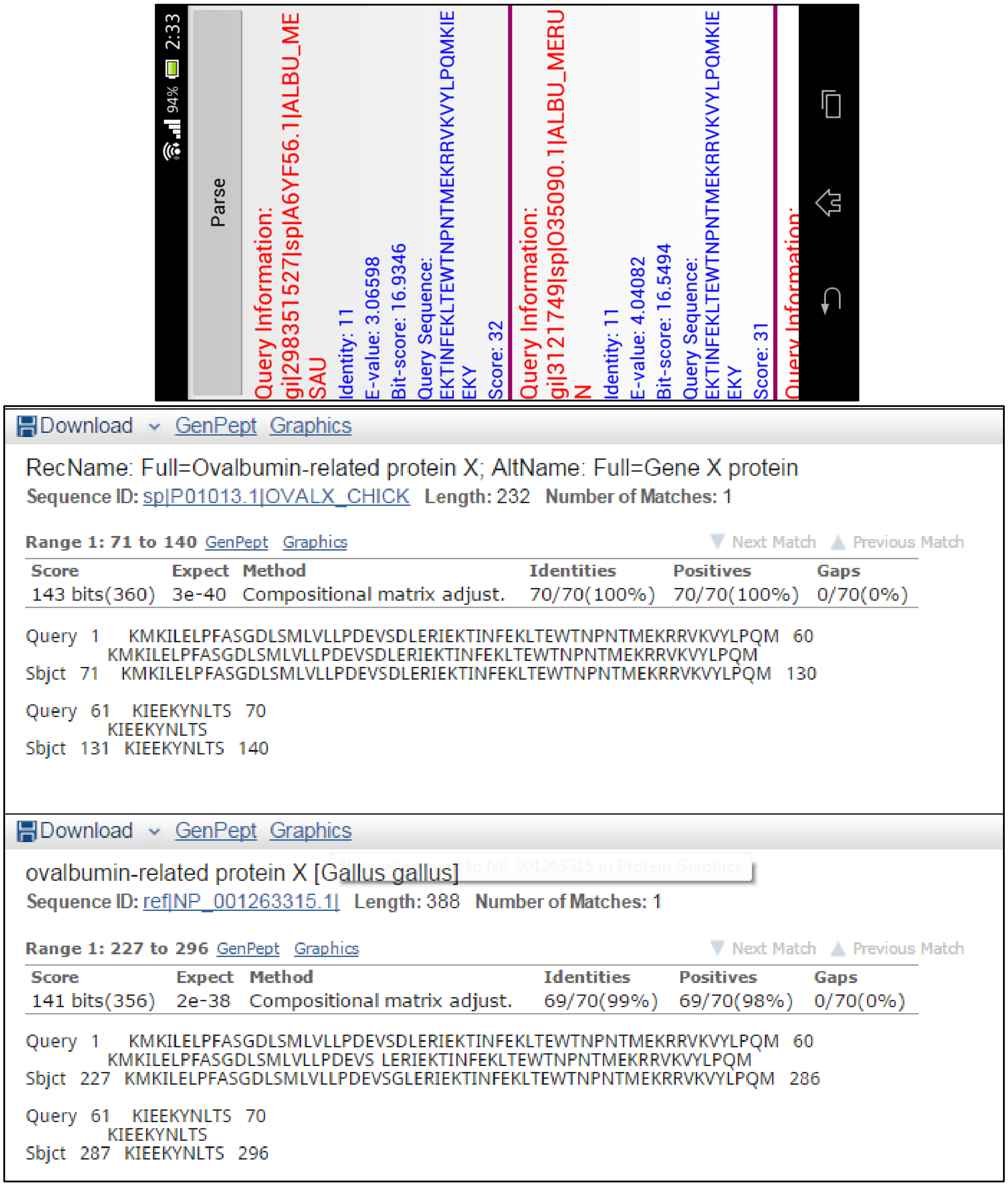
Score	Expect	Method	Identities	Positives	Gaps
148 bits(373)	9e-41	Compositional matrix adjust.	69/70(99%)	70/70(100%)	0/70(0%)

```
Query  1    QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMMCMNN  60
            QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTK+ESKPVQMMCMNN
Sbjct  157  QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKEESKPVQMMCMNN  216

Query  61   SFNVATLPAE  70
            SFNVATLPAE
Sbjct  217  SFNVATLPAE  226
```

Fig. 3.16. Comparison of results for the same given input.

Sequence:

KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKV YLPQMKIEEKYNLTS



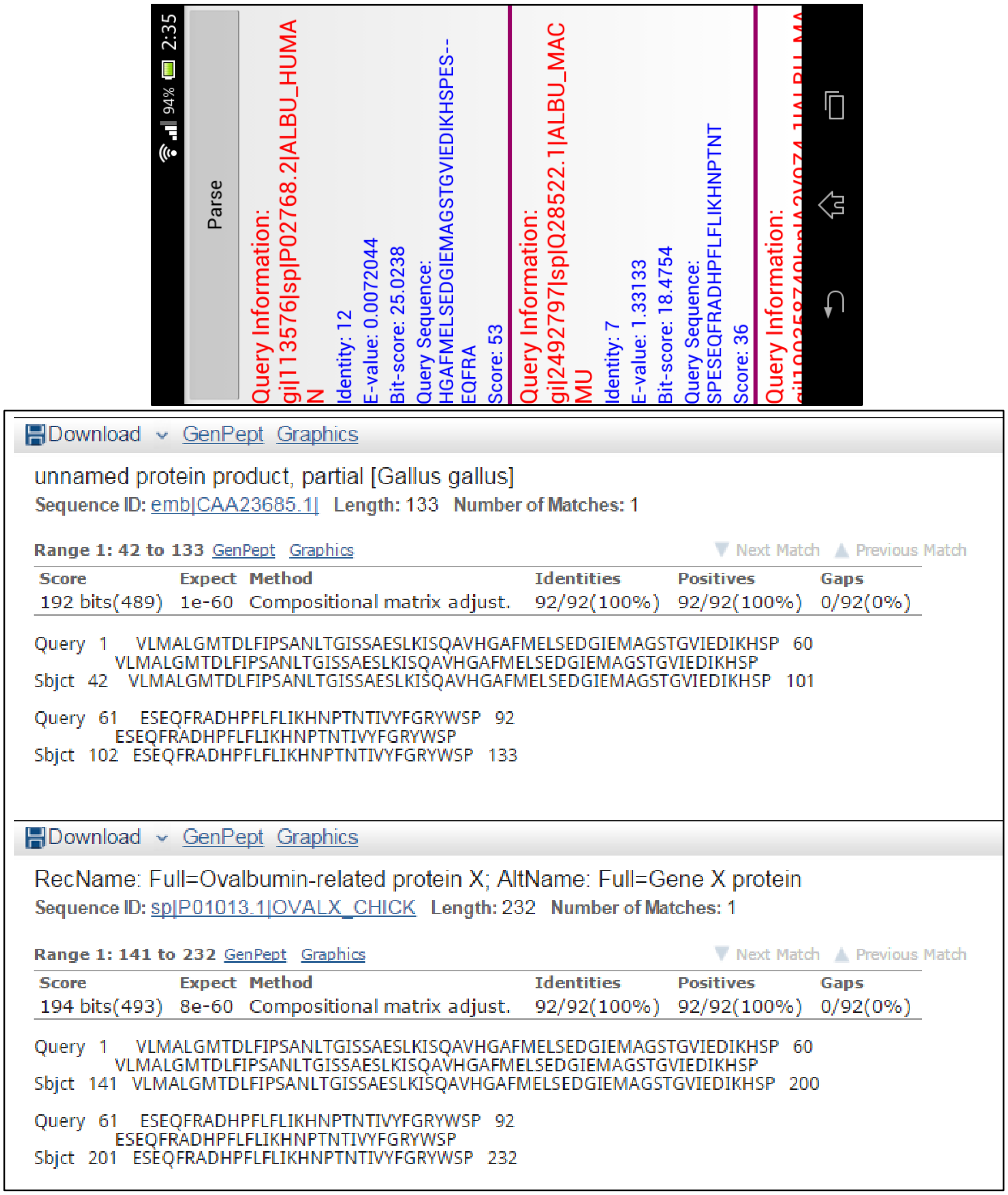
```
2:33  94%
Parse

Query Information:
gi|298351527|sp|A6YF56.1|ALBU_MESAU
Identity: 11
E-value: 3.06598
Bit-score: 16.9346
Query Sequence:
EKTINFEKLTEWTNPNTMEKRRVKVYLPQMKIEEKY
Score: 32

Query Information:
gi|3121749|sp|O35090.1|ALBU_MERUN
Identity: 11
E-value: 4.04082
Bit-score: 16.5494
Query Sequence:
EKTINFEKLTEWTNPNTMEKRRVKVYLPQMKIEEKY
Score: 31

Query Information:
```

```
Download ⌄  GenPept  Graphics

RecName: Full=Ovalbumin-related protein X; AltName: Full=Gene X protein
Sequence ID: sp|P01013.1|OVALX_CHICK  Length: 232  Number of Matches: 1

Range 1: 71 to 140  GenPept  Graphics                    ▼ Next Match  ▲ Previous Match
 Score           Expect  Method                         Identities    Positives     Gaps
 143 bits(360)   3e-40   Compositional matrix adjust.   70/70(100%)   70/70(100%)   0/70(0%)

Query  1    KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQM  60
            KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQM
Sbjct  71   KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQM  130

Query  61   KIEEKYNLTS  70
            KIEEKYNLTS
Sbjct  131  KIEEKYNLTS  140

Download ⌄  GenPept  Graphics

ovalbumin-related protein X [Gallus gallus]
Sequence ID: ref|NP_001263315.1|  Length: 388  Number of Matches: 1

Range 1: 227 to 296  GenPept  Graphics                  ▼ Next Match  ▲ Previous Match
 Score           Expect  Method                         Identities    Positives     Gaps
 141 bits(356)   2e-38   Compositional matrix adjust.   69/70(99%)    69/70(98%)    0/70(0%)

Query  1    KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQM  60
            KMKILELPFASGDLSMLVLLPDEVS LERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQM
Sbjct  227  KMKILELPFASGDLSMLVLLPDEVSGLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQM  286

Query  61   KIEEKYNLTS  70
            KIEEKYNLTS
Sbjct  287  KIEEKYNLTS  296
```

Fig. 3.17. Comparison of results for the same given input.

Sequence:

VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDI KHSPESEQFRADHPFLFLIKHNPTNTIVYFGRYWSP



Query Information:
gi|113576|sp|P02768.2|ALBU_HUMAN
Identity: 12
E-value: 0.0072044
Bit-score: 25.0238
Query Sequence:
HGAFMELSEDGIEMAGSTGVIEDIKHSPES--EQFRA
Score: 53

Query Information:
gi|2492797|sp|Q28522.1|ALBU_MACMU
Identity: 7
E-value: 1.33133
Bit-score: 18.4754
Query Sequence:
SPESEQFRADHPFLFLIKHNPTNT
Score: 36

Query Information:

unnamed protein product, partial [Gallus gallus]
Sequence ID: emb|CAA23685.1| **Length:** 133 **Number of Matches:** 1

Range 1: 42 to 133 GenPept Graphics

Score	Expect	Method	Identities	Positives	Gaps
192 bits(489)	1e-60	Compositional matrix adjust.	92/92(100%)	92/92(100%)	0/92(0%)

```
Query  1    VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSP  60
            VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSP
Sbjct  42   VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSP  101

Query  61   ESEQFRADHPFLFLIKHNPTNTIVYFGRYWSP  92
            ESEQFRADHPFLFLIKHNPTNTIVYFGRYWSP
Sbjct  102  ESEQFRADHPFLFLIKHNPTNTIVYFGRYWSP  133
```

RecName: Full=Ovalbumin-related protein X; AltName: Full=Gene X protein
Sequence ID: sp|P01013.1|OVALX_CHICK **Length:** 232 **Number of Matches:** 1

Range 1: 141 to 232 GenPept Graphics

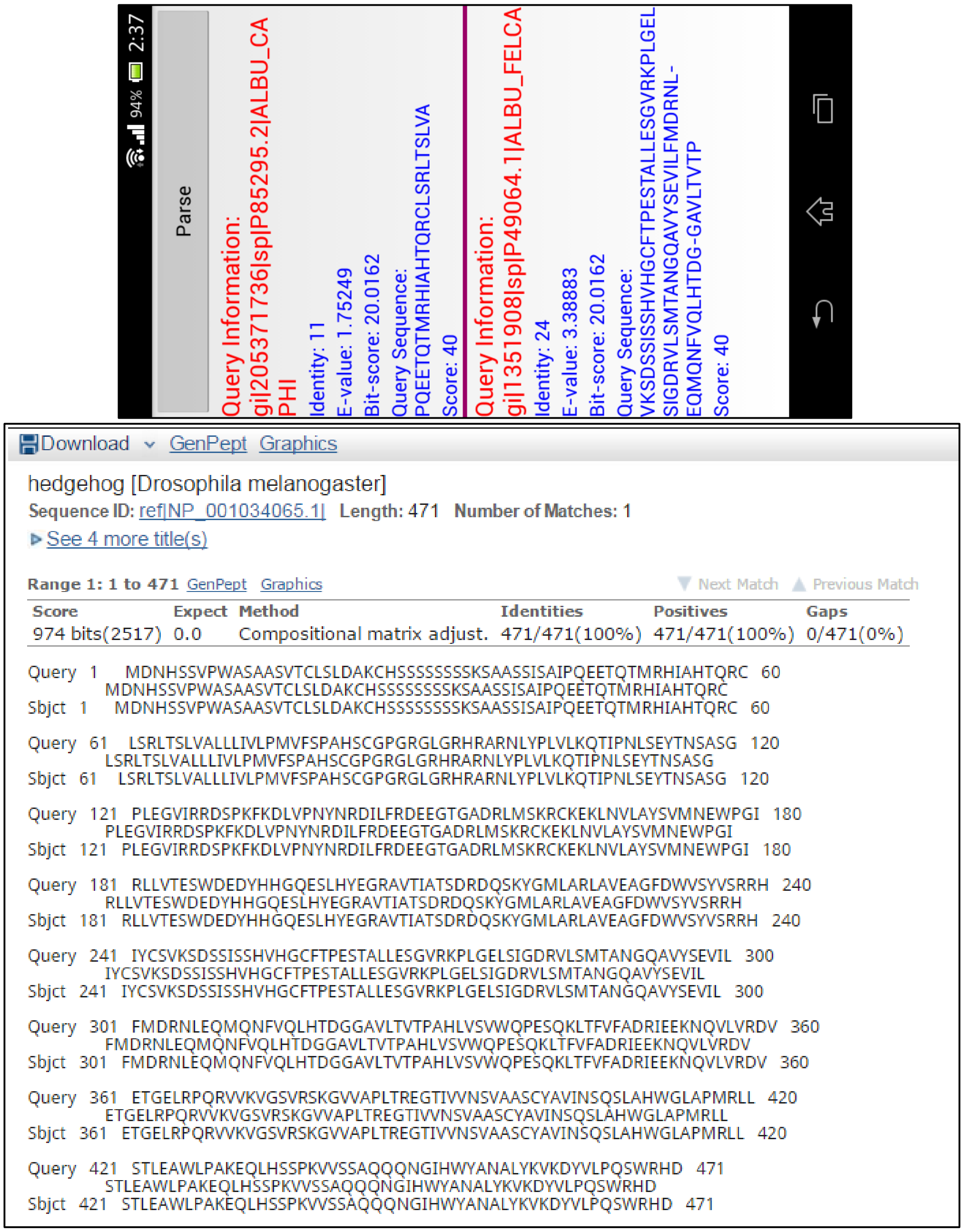
Score	Expect	Method	Identities	Positives	Gaps
194 bits(493)	8e-60	Compositional matrix adjust.	92/92(100%)	92/92(100%)	0/92(0%)

```
Query  1    VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSP  60
            VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSP
Sbjct  141  VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSP  200

Query  61   ESEQFRADHPFLFLIKHNPTNTIVYFGRYWSP  92
            ESEQFRADHPFLFLIKHNPTNTIVYFGRYWSP
Sbjct  201  ESEQFRADHPFLFLIKHNPTNTIVYFGRYWSP  232
```

Fig. 3.18. Comparison of results for the same given input.

Accession ID:

NP_001034065.1



Query Information:
gi|205371736|sp|P85295.2|ALBU_CAPHI
Identity: 11
E-value: 1.75249
Bit-score: 20.0162
Query Sequence:
PQEETQTMRHIAHTQRCLSRLTSLVA
Score: 40

Query Information:
gi|1351908|sp|P49064.1|ALBU_FELCA
Identity: 24
E-value: 3.38883
Bit-score: 20.0162
Query Sequence:
VKSDSSISSHVHGCFTPESTALLESGVRKPLGELSIGDRVLSMTANGQAVYSEVILFMDRNL-EQMQNFVQLHTDG-GAVLTVTP
Score: 40

hedgehog [Drosophila melanogaster]
Sequence ID: ref|NP_001034065.1| Length: 471 Number of Matches: 1
See 4 more title(s)

Range 1: 1 to 471 GenPept Graphics

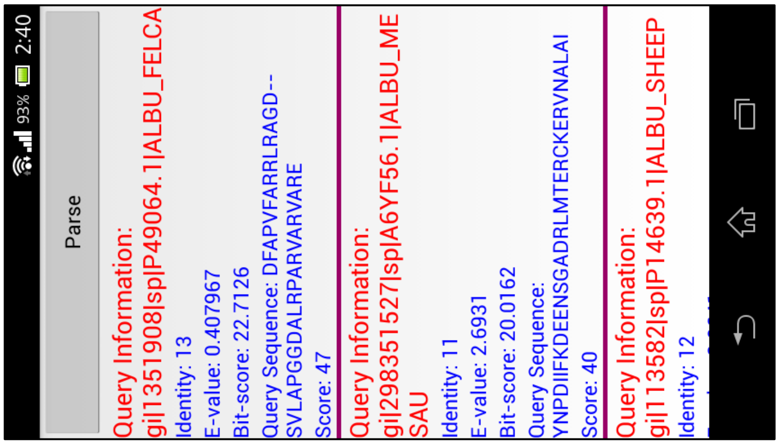
Score	Expect	Method	Identities	Positives	Gaps
974 bits(2517)	0.0	Compositional matrix adjust.	471/471(100%)	471/471(100%)	0/471(0%)

```
Query  1    MDNHSSVPWASAASVTCLSLDAKCHSSSSSSSSKSAASSISAIPQEETQTMRHIAHTQRC  60
            MDNHSSVPWASAASVTCLSLDAKCHSSSSSSSSKSAASSISAIPQEETQTMRHIAHTQRC
Sbjct  1    MDNHSSVPWASAASVTCLSLDAKCHSSSSSSSSKSAASSISAIPQEETQTMRHIAHTQRC  60

Query  61   LSRLTSLVALLLIVLPMVFSPAHSCGPGRGLGRHRARNLYPLVLKQTIPNLSEYTNSASG  120
            LSRLTSLVALLLIVLPMVFSPAHSCGPGRGLGRHRARNLYPLVLKQTIPNLSEYTNSASG
Sbjct  61   LSRLTSLVALLLIVLPMVFSPAHSCGPGRGLGRHRARNLYPLVLKQTIPNLSEYTNSASG  120

Query  121  PLEGVIRRDSPKFKDLVPNYNRDILFRDEEGTGADRLMSKRCKEKLNVLAYSVMNEWPGI  180
            PLEGVIRRDSPKFKDLVPNYNRDILFRDEEGTGADRLMSKRCKEKLNVLAYSVMNEWPGI
Sbjct  121  PLEGVIRRDSPKFKDLVPNYNRDILFRDEEGTGADRLMSKRCKEKLNVLAYSVMNEWPGI  180

Query  181  RLLVTESWDEDYHHGQESLHYEGRAVTIATSDRDQSKYGMLARLAVEAGFDWVSYVSRRH  240
            RLLVTESWDEDYHHGQESLHYEGRAVTIATSDRDQSKYGMLARLAVEAGFDWVSYVSRRH
Sbjct  181  RLLVTESWDEDYHHGQESLHYEGRAVTIATSDRDQSKYGMLARLAVEAGFDWVSYVSRRH  240

Query  241  IYCSVKSDSSISSHVHGCFTPESTALLESGVRKPLGELSIGDRVLSMTANGQAVYSEVIL  300
            IYCSVKSDSSISSHVHGCFTPESTALLESGVRKPLGELSIGDRVLSMTANGQAVYSEVIL
Sbjct  241  IYCSVKSDSSISSHVHGCFTPESTALLESGVRKPLGELSIGDRVLSMTANGQAVYSEVIL  300

Query  301  FMDRNLEQMQNFVQLHTDGGAVLTVTPAHLVSVWQPESQKLTFVFADRIEEKNQVLVRDV  360
            FMDRNLEQMQNFVQLHTDGGAVLTVTPAHLVSVWQPESQKLTFVFADRIEEKNQVLVRDV
Sbjct  301  FMDRNLEQMQNFVQLHTDGGAVLTVTPAHLVSVWQPESQKLTFVFADRIEEKNQVLVRDV  360

Query  361  ETGELRPQRVVKVGSVRSKGVVAPLTREGTIVVNSVAASCYAVINSQSLAHWGLAPMRLL  420
            ETGELRPQRVVKVGSVRSKGVVAPLTREGTIVVNSVAASCYAVINSQSLAHWGLAPMRLL
Sbjct  361  ETGELRPQRVVKVGSVRSKGVVAPLTREGTIVVNSVAASCYAVINSQSLAHWGLAPMRLL  420

Query  421  STLEAWLPAKEQLHSSPKVVSSAQQQNGIHWYANALYKVKDYVLPQSWRHD  471
            STLEAWLPAKEQLHSSPKVVSSAQQQNGIHWYANALYKVKDYVLPQSWRHD
Sbjct  421  STLEAWLPAKEQLHSSPKVVSSAQQQNGIHWYANALYKVKDYVLPQSWRHD  471
```

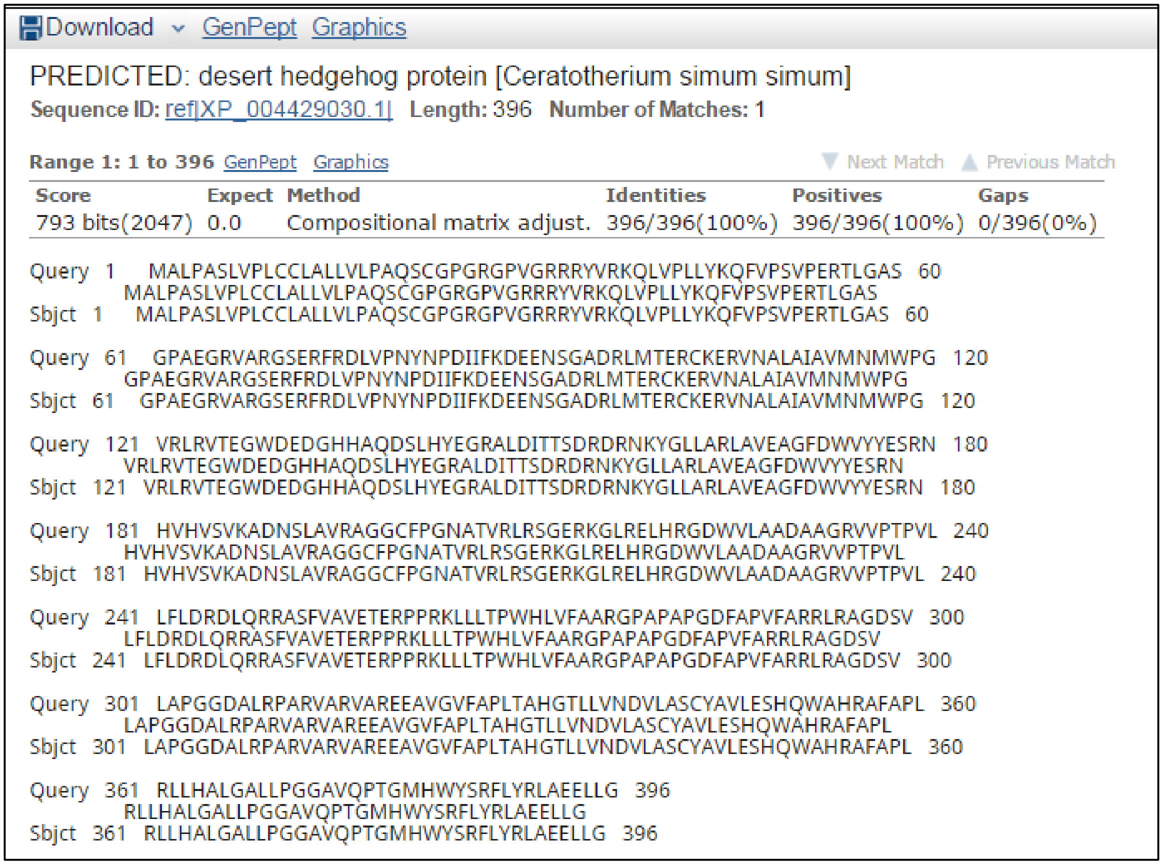
Fig. 3.19. Comparison of results for the same given input.

Accession ID:

XP_004429030.1



Query Information:
gi|1351908|sp|P49064.1|ALBU_FELCA
Identity: 13
E-value: 0.407967
Bit-score: 22.7126
Query Sequence: DFAPVFARRLRAGD--SVLAPGGDALRPARVARVARE
Score: 47
Query Information:
gi|298351527|sp|A6YF56.1|ALBU_MESAU
Identity: 11
E-value: 2.6931
Bit-score: 20.0162
Query Sequence: YNPDIIFKDEENSGADRLMTERCKERVNALAI
Score: 40
Query Information:
gi|113582|sp|P14639.1|ALBU_SHEEP
Identity: 12



PREDICTED: desert hedgehog protein [Ceratotherium simum simum]
Sequence ID: ref|XP_004429030.1| Length: 396 Number of Matches: 1

Range 1: 1 to 396 GenPept Graphics ▼ Next Match ▲ Previous Match

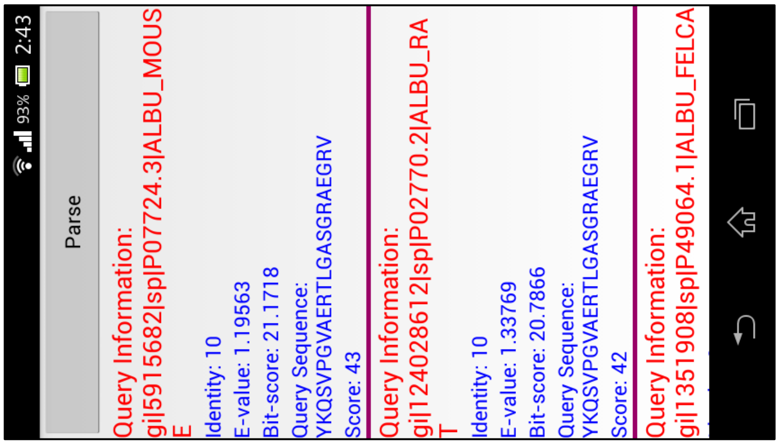
Score	Expect	Method	Identities	Positives	Gaps
793 bits(2047)	0.0	Compositional matrix adjust.	396/396(100%)	396/396(100%)	0/396(0%)

```
Query  1    MALPASLVPLCCLALLVLPAQSCGPGRGPVGRRRYVRKQLVPLLYKQFVPSVPERTLGAS  60
            MALPASLVPLCCLALLVLPAQSCGPGRGPVGRRRYVRKQLVPLLYKQFVPSVPERTLGAS
Sbjct  1    MALPASLVPLCCLALLVLPAQSCGPGRGPVGRRRYVRKQLVPLLYKQFVPSVPERTLGAS  60

Query  61   GPAEGRVARGSERFRDLVPNYNPDIIFKDEENSGADRLMTERCKERVNALAIAVMNMWPG  120
            GPAEGRVARGSERFRDLVPNYNPDIIFKDEENSGADRLMTERCKERVNALAIAVMNMWPG
Sbjct  61   GPAEGRVARGSERFRDLVPNYNPDIIFKDEENSGADRLMTERCKERVNALAIAVMNMWPG  120

Query  121  VRLRVTEGWDEDGHHAQDSLHYEGRALDITTSDRDRNKYGLLARLAVEAGFDWVYYESRN  180
            VRLRVTEGWDEDGHHAQDSLHYEGRALDITTSDRDRNKYGLLARLAVEAGFDWVYYESRN
Sbjct  121  VRLRVTEGWDEDGHHAQDSLHYEGRALDITTSDRDRNKYGLLARLAVEAGFDWVYYESRN  180

Query  181  HVHVSVKADNSLAVRAGGCFPGNATVRLRSGERKGLRELHRGDWVLAADAAGRVVPTPVL  240
            HVHVSVKADNSLAVRAGGCFPGNATVRLRSGERKGLRELHRGDWVLAADAAGRVVPTPVL
Sbjct  181  HVHVSVKADNSLAVRAGGCFPGNATVRLRSGERKGLRELHRGDWVLAADAAGRVVPTPVL  240

Query  241  LFLDRDLQRRASFVAVETERPPRKLLLTPWHLVFAARGPAPAPGDFAPVFARRLRAGDSV  300
            LFLDRDLQRRASFVAVETERPPRKLLLTPWHLVFAARGPAPAPGDFAPVFARRLRAGDSV
Sbjct  241  LFLDRDLQRRASFVAVETERPPRKLLLTPWHLVFAARGPAPAPGDFAPVFARRLRAGDSV  300

Query  301  LAPGGDALRPARVARVAREEAVGVFAPLTAHGTLLVNDVLASCYAVLESHQWAHRAFAPL  360
            LAPGGDALRPARVARVAREEAVGVFAPLTAHGTLLVNDVLASCYAVLESHQWAHRAFAPL
Sbjct  301  LAPGGDALRPARVARVAREEAVGVFAPLTAHGTLLVNDVLASCYAVLESHQWAHRAFAPL  360

Query  361  RLLHALGALLPGGAVQPTGMHWYSRFLYRLAEELLG  396
            RLLHALGALLPGGAVQPTGMHWYSRFLYRLAEELLG
Sbjct  361  RLLHALGALLPGGAVQPTGMHWYSRFLYRLAEELLG  396
```

Fig. 3.20. Comparison of results for the same given input.

Accession ID:

KQL88370.1



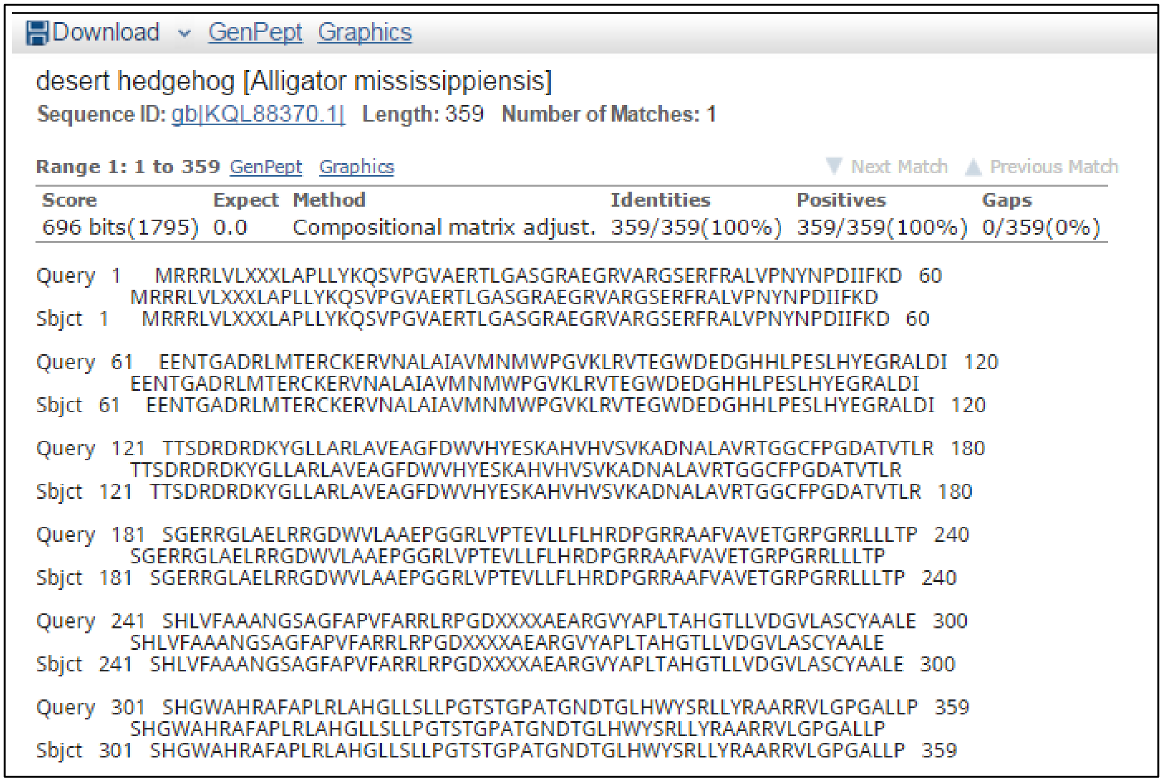
Query Information:
gi|59155682|sp|P07724.3|ALBU_MOUSE

Identity: 10
E-value: 1.19563
Bit-score: 21.1718
Query Sequence:
YKQSVPGVAERTLGASGRAEGRV
Score: 43

Query Information:
gi|124028612|sp|P02770.2|ALBU_RAT

Identity: 10
E-value: 1.33769
Bit-score: 20.7866
Query Sequence:
YKQSVPGVAERTLGASGRAEGRV
Score: 42

Query Information:
gi|1351908|sp|P49064.1|ALBU_FELCA



desert hedgehog [Alligator mississippiensis]

Sequence ID: gb|KQL88370.1| Length: 359 Number of Matches: 1

Range 1: 1 to 359 GenPept Graphics

Score	Expect	Method	Identities	Positives	Gaps
696 bits(1795)	0.0	Compositional matrix adjust.	359/359(100%)	359/359(100%)	0/359(0%)

```
Query  1    MRRRLVLXXXLAPLLYKQSVPGVAERTLGASGRAEGRVARGSERFRALVPNYNPDIIFKD  60
            MRRRLVLXXXLAPLLYKQSVPGVAERTLGASGRAEGRVARGSERFRALVPNYNPDIIFKD
Sbjct  1    MRRRLVLXXXLAPLLYKQSVPGVAERTLGASGRAEGRVARGSERFRALVPNYNPDIIFKD  60

Query  61   EENTGADRLMTERCKERVNALAIAVMNMWPGVKLRVTEGWDEDGHHLPESLHYEGRALDI  120
            EENTGADRLMTERCKERVNALAIAVMNMWPGVKLRVTEGWDEDGHHLPESLHYEGRALDI
Sbjct  61   EENTGADRLMTERCKERVNALAIAVMNMWPGVKLRVTEGWDEDGHHLPESLHYEGRALDI  120

Query  121  TTSDRDRDKYGLLARLAVEAGFDWVHYESKAHVHVSVKADNALAVRTGGCFPGDATVTLR  180
            TTSDRDRDKYGLLARLAVEAGFDWVHYESKAHVHVSVKADNALAVRTGGCFPGDATVTLR
Sbjct  121  TTSDRDRDKYGLLARLAVEAGFDWVHYESKAHVHVSVKADNALAVRTGGCFPGDATVTLR  180

Query  181  SGERRGLAELRRGDWVLAAEPGGRLVPTEVLLFLHRDPGRRAAFVAVETGRPGRRLLLTP  240
            SGERRGLAELRRGDWVLAAEPGGRLVPTEVLLFLHRDPGRRAAFVAVETGRPGRRLLLTP
Sbjct  181  SGERRGLAELRRGDWVLAAEPGGRLVPTEVLLFLHRDPGRRAAFVAVETGRPGRRLLLTP  240

Query  241  SHLVFAAANGSAGFAPVFARRLRPGDXXXXAEARGVYAPLTAHGTLLVDGVLASCYAALE  300
            SHLVFAAANGSAGFAPVFARRLRPGDXXXXAEARGVYAPLTAHGTLLVDGVLASCYAALE
Sbjct  241  SHLVFAAANGSAGFAPVFARRLRPGDXXXXAEARGVYAPLTAHGTLLVDGVLASCYAALE  300

Query  301  SHGWAHRAFAPLRLAHGLLSLLPGTSTGPATGNDTGLHWYSRLLYRAARRVLGPGALLP  359
            SHGWAHRAFAPLRLAHGLLSLLPGTSTGPATGNDTGLHWYSRLLYRAARRVLGPGALLP
Sbjct  301  SHGWAHRAFAPLRLAHGLLSLLPGTSTGPATGNDTGLHWYSRLLYRAARRVLGPGALLP  359
```

Fig. 3.21. Comparison of results for the same given input.

CHAPTER IV: Conclusion

4. CONCLUSION

Everything changes with time. With the disciplined flow of time, science has also progressed with similar restraint. It seems not too long ago, in 1998, Wheaton College (Norton, MA) has actively engaged with various modes of interdisciplinary teaching [13]. In fact, collaborative pedagogy in the new curriculum is centered on connections, or pairs of linked courses that connect significantly different disciplines. The hybridization of biology and information technology came long after that, yet the impact of the bioinformatics field in our lives is so profound. The growth of biotechnology research has yielded a huge database of vast diverse database. Utilizing this database and keeping track of it, initially led to bioinformatics. Nowadays data capture, data warehousing and data mining have become major issues for biotechnologists and biological scientists due to sudden growth in quantitative data in biology, such as, complete genomes of biological species, including human genome, protein sequences, protein 3-dimensional structures, metabolic pathway databases, cell lines, biodiversity related information. Advancement in information technology, particularly Internet is being used to gather, access and disseminate the ever- increasing information in biology and biotechnology. In order to keep pace with the advancing technology it is required to incorporate the available information to all fronts of technological interface available to the world.

The leading interface that reaches throughout a vast population of the globe is undoubtedly smartphones. As per it can be easily grasped how crucial it is to incorporate the available databases to a smartphone platform. To make a smartphone based application, firstly features needed to be selected and a clear concept of what the application would be like was required. In order to bring this goal to life strategic steps had to be taken. Initial target was to find the optimal algorithm for DNA or protein sequence alignment and make sure it would work smoothly in a limited resource environment. Tests and comparisons in these two criteria led us to the best algorithm suited to the application, which is BLAST. The next step was to implement the BLAST algorithm in a suitable language. For this implementation Java language was mostly used. The BLAST algorithm was then incorporated into an application for smartphones. In order to perform sequence alignment the application needed to be connected to the server containing the database full of known and partially known protein sequences. This very crucial connection to the server was made using XAMPP. Another very important feature, the local alignment was implemented in a similar manner. Firstly the Smith-Waterman algorithm was implemented in the language java, then it was incorporated to the application as a feature. Blosum62 matrix was used to calculate the score. Many difficulties were faced while implementing these steps. One significant challenge faced was the server delay and server timing out. Data parsing to the main algorithm had to be revised thoroughly to check this problem. After the main features were up and running, the interface of the application was made as eye pleasing and user friendly as possible. This incredibly convenient application would help researchers progress so much faster. No one likes to carry their laptops all the time, here is where the application comes in. As it is not feasible to carry a laptop around all the time, the features were put so that you may require in something that you do carry all the time. This application provides all the vital information needed for research or study such as e-value, accession ID, scientific name of the organism. With the advancement of technology the processing capabilities and storage capabilities of smartphones has increased almost exponentially. Almost everything can be put on a smartphone one day though already it has advanced to an ultimate level.

4.1 Future plans

This mobile application has some evolution ground to make in near future. Lots of other features are expected to be included in the near future. Some of the features that would give this application a major boost in the market are described below.

4.1.1 Different platforms

For this particular project only Android implemented smartphones were considered. After the success, some other platforms are proposed to be tried on. Some of these platforms are Windows, iOS.

4.1.2 Login feature

The very first feature that is wished to be included in this mobile application is a user login system. This feature will help the user to have a personal record of the searching s/he has done over the time in this application. This feature will keep separate information of the many different user in the database of the application administrator.

4.1.3 Saving data to cloud

The edition of this feature would be a great help for the researchers that wish to save their findings in the cloud so that they can use them later when they are on a computer. This will give them the opportunity to save on the cloud.

CHAPTER V: References

5. REFERENCES

[1] R. Shrestha, 'The Importance of Biotechnology in Today's Time', *Aakhayan a chapter scripting life*, 2015. .

[2] Biotechonweb.com, 'Applications of Biotech In Medical : Projects In Biopharmaceutical', 2015. [Online]. Available: http://www.biotechonweb.com/Application-of-biotech-in-Medical.html. [Accessed: 09- Dec- 2015].

[3] Biotechlearn.org.nz, 'Forensics | Biotech Learning Hub', 2015. [Online]. Available: http://biotechlearn.org.nz/focus_stories/forensics. [Accessed: 09- Dec- 2015].

[4] Informatics.sdsu.edu, 'Biological and Medical Informatics Research Center (BMIRC) – Bioinformatics', 2015. [Online]. Available: http://informatics.sdsu.edu/bioinformatics/. [Accessed: 09- Dec- 2015].

[5] 'The Growing Importance of Smart Phones', *Idea File*, pp. 1-8, 2011.

[6] M. Fourment and M. Gillings, 'A comparison of common programming languages used in bioinformatics', *BMC Bioinformatics*, vol. 9, no. 1, p. 82, 2008.

[7] S. Shavor, *The Java developer's guide to Eclipse*. Boston, MA: Addison-Wesley, 2003.

[8] A. Chan, *An Analysis of Pairwise Sequence Alignment Algorithm Complexities: Needleman-Wunsch, Smith-Waterman, FASTA, BLAST and Gapped BLAST*, 1st ed. 2015, p. 12.

[9] G. Vej, *Bioinformatics explained: BLAST versus Smith-Waterman*, 1st ed. Aarhus: www.clcbio.com, 2007, p. 7.

[10] A. Lew and H. Mauch, *Dynamic programming*. Berlin: Springer, 2007.

[11] M. Epelman, *Introduction to Integer Programming*, 1st ed. 2012, pp. 111-132.

[12] M. Bermingham, R. Pong-Wong, A. Spiliopoulou, C. Hayward, I. Rudan, H. Campbell, A. Wright, J. Wilson, F. Agakov, P. Navarro and C. Haley, 'Application of high-dimensional feature selection: evaluation for genomic prediction in man', *Sci. Rep.*, vol. 5, p. 10312, 2015.

[13] M. Maloney, J. Parker, M. LeBlanc, C. Woodard, M. Glackin and M. Hanrahan, 'Bioinformatics and the Undergraduate Curriculum', *Cell Biology Education*, vol. 9, no. 3, pp. 172-174, 2010.

[14] D. Mount, "Using the Basic Local Alignment Search Tool (BLAST)", *Cold Spring Harbor Protocols*, vol. 2007, no. 14, pp. pdb.top17-pdb.top17, 2007.

[15] Ncbi.nlm.nih.gov, "NCBI News | Summer 1999", 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/Web/Newsltr/Summer99/qblast.html. [Accessed: 20-Dec- 2015].