Bioinformatics: Analyzing DNA Sequence using BLAST

By

Nadim Naimur Rahman
ID#03201042

Department of Computer Science and Engineering
BRAC University

Thesis submitted to the faculty of the
BRAC University
In partial fulfillment of the requirements of the degree of
BACHELOR OF SCIENCE
in
Computer Science and Engineering

Supervised By

Dr. Mumit Khan
Associate Professor
BRAC University

September 5th, 2007
BRAC University
DHAKA

DECLARATION

We, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by me under the supervision of Dr. Mumit Khan, Associate Professor, Department of Computer Science and Engineering, BRAC University, Dhaka. I also declare that no part of this thesis and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Sign

(Nadim Naimur Rahman)

Countersigned

(Dr. Mumit Khan)
Supervisor

Acknowledgements:

I would like to thank my thesis supervisor Dr. Mumit Khan for his all- out help and cooperation during the period of my thesis. His guidance played a pivotal part in completing my thesis.

I am also thankful to Dr. Naiyyum Chowdhury and Ms. Nazli Sharmin of the BRAC University Biotechnology Department for helping me out in the interpretation process of the output.

Bioinformatics: Analyzing DNA Sequence using BLAST

Nadim Naimur Rahman

Abstract

This paper attempts to use the BLAST simulator to analyze a DNA sequence and interpret the results in a way that are understandable for biotechnologists. It shows how to install, build and run the simulator using an input DNA sequence, comparing it with a database and obtain an output that can be used for many different purposes.

The paper also discusses the areas where the results can be put to use and how it can be utilized for the benefit of the people of Bangladesh.

Table of Contents:

1.0. INTRODUCTION:

Bioinformatics is a buzzword in this modern era of scientific research. A lot of work has started in this field. It all started way back in 1990 under the name Human Genome Project (HGP) at the National Institutes of Health (NIH) in the United States. The main aim of the project was to understand the genetic makeup of the human species by identifying all the genes in the human genome and mapping how individual genes are sequenced. By some definitions the HGP was completed in 2005.

Technically, it can be said that human genes consist of blocks of amino acids or proteins. So, the main challenge of bioinformatics was to analyze these protein sequences and derive information that can be put to use for different purposes. This paper also discusses some scopes of bioinformatics and how they can be interlinked with medical science.

However, the main aim of this paper is to show the efficient usage of the BLAST simulator. It shows the installation, usage and the interpretation process of the simulator in detail so that the derived information can be used for further use in different appropriate areas.

1.1. Scope of Bioinformatics:

It can be easily said that medical science would be the biggest beneficiary from the research carried out on bioinformatics. The core foundation of bioinformatics lies from the fact that virtually all medical conditions have a genetic component. Therefore, if physicians can use the genetic information of the patient in order to properly diagnose and treat the disease.

Other than that, one huge benefit patients can get from genetic treatment is the use of individualized drugs. Physicians would be able to prescribe them with drugs aiming to treat the component of the genes that is the root cause of the disease.

It may be mentioned that there are some diseases like cancer that are caused by mutation of a single gene or by the interaction of many genes. These genes are termed as risk factors by the physicians. Using genetic information, physicians would be able to identify these risk factors and administer genetic treatment like gene therapy to the patients. It has been experimentally seen so far that gene therapy is a far more efficient treatment process for treating cancer at the initial stages. The speed of recovery for the patient is much faster when compared with chemotherapy and the treatment is also far less stressful for the patient.

1.2. Explanation of a sequence:

This research was conducted on DNA sequences only where DNA stands for Deoxyribonucleic Acid. It consists of a double helix pattern known as strands and is made up of long chains of amino acids which are infact proteins. However, for simplicity of the research, only a single strand is used.

A DNA sequence can be thought of as a language consisting of four alphabets. They are:

1. Adenine (A)
2. Cytosine (C)
3. Guanine (G)
4. Thymine (T)

Therefore, it can be implied that a DNA sequence will always be a language of four characters and is commonly known as the “Book of Life”.

1.3. Challenges in Bioinformatics:

Bioinformatics can be termed as a melting pot of many fields such as computer science, statistics, biology, chemistry, mathematics, etc. Therefore, an integrated approach was required to conduct research in this field. However, the main challenge in bioinformatics was sequence alignment. It may be mentioned that protein sequences can be of infinite length. Therefore, sequence alignment required tools that can match a given input sequence with that stored in a database. In short, a method needed to be devised where the computer will understand the inputs, match them and give outputs that can be interpreted by biologists to derive information. This is where BLAST can play an instrumental role.

By definition, bioinformatics is: ***"The mathematical, statistical and computing methods that aim to solve biological problems using DNA and amino acid sequences and related information."***

2.0. BASIC LOCAL ALIGNMENT SEARCH TOOL

2.1 BLAST:

BLAST stands for Basic Local Alignment Search Tool. It was developed by the National Centre for Biotechnology Information (NCBI) in USA. Even though the name of BLAST suggests only local alignment, but in reality it can carry out both local and global alignment. The main idea behind BLAST was to feed the computer with an input sequence and produce an output when it is run against a selected database. The output is then interpreted to derive information about the input sequence.

In order to do this, the most important thing BLAST had to carry out is string matching, since it had to align sequences which, at times, might get as long as thousands of characters in length. However, the string matching algorithms used by BLAST will not be discussed here as it goes beyond the scope of this paper. This paper will only focus on the outputs produced by BLAST and their sample interpretations.

Lastly, it may be mentioned that BLAST is currently the most widely used sequence alignment tool used in the most renowned universities of the world conducting research on bioinformatics.

2.2 Research Methodology:

The following steps were carried out while conducting the research.

1. **Downloading the simulator**: In this step the BLAST simulator was downloaded from the NCBI website.
2. **Downloading the database**: Five databases were downloaded from the NCBI website for conducting the research.
3. **Building BLAST**: The software was built and the database was connected to the software so that simulations can be carried out.
4. **Testing a Sequence**: A test sequence was entered, the simulator was run successfully and an output was obtained.
5. **Interpretation of the output**: The output was analyzed and related information was derived from the database.

However, in parallel to the above- mentioned steps, extensive academic studies were done on bioinformatics, algorithms and the usage of BLAST from different books, journals and papers.

2.3 Local Alignment:

As mentioned earlier, the main purpose of using BLAST is sequence alignment. In carrying out a local alignment, BLAST breaks down an input sequence into smaller parts and compares them with the database. No gaps are introduced in local alignment in order to force the input sequence to match with the database. A match or a mismatch is indicated by the presence or absence of a vertical line between the alphabets of the input sequence and the database sequence. Two sample examples of local alignment are given below:

Local Alignment Example-1:

```
Input:  ATTGCTTCTAGGA
        |||||||||||||
Query:  ATTGCTTCTAGGA
```

Expect= 0.74

Identities= 13/13 (100%)

Strand= Plus/ Plus

Note: No mismatch.

Input: Defines the input sequence.

Query: The sequence matched from the database.

Expect: The probability that a sequence will match by chance. The less this value, the better is for analysis.

Identities: Shows the percentage of matching in the sequence.

Strand: Plus/Plus means that another protein sequence can be obtained by the interaction of these two sequences. Plus/Minus means that a protein sequence cannot be obtained by the interaction of these two sequences.

Note: Vertical lines indicate the matches.

Local Alignment Example-2:

```
Input:  ATTGCTTCTAGGA
        |||||| ||||||
Query:  ATTGCTCCTAGGA
```

Expect= 0.74

Identities= 12/13 (92%)

Strand= Plus/ Plus

Note: Mismatch found by the absence of a vertical line between T and C.

Please refer to the appendix for the complete output of the sequence analysis.

2.4. Global Alignment:

Unlike local alignment, sequences are guaranteed to match with the input sequence. Here, gaps are introduced whenever a mismatch is found and the input sequence is shifted by some places in order to match it with the database. One other important aspect of global alignment is that the whole sequence is taken at once and matched with the database, unlike in local alignment where an input sequence is broken down into smaller components. An example of global alignment is shown below:

Global Alignment Example:

```
Input: ATT         G
       |||         |
Query: ATTAAAAAAAAAG
```

Expect= 0.74

Identities= 13/13 (100%)

Strand= Plus/ Plus

Note: Gaps introduced for the sequences to match.

Please refer to the appendix for the complete output of the sequence analysis.

3.0 PURPOSE AND FUTURE WORKS

3.1. Future Improvements:

So far the main focus of the research was at a system level. It mainly included the building, use and obtaining the output of BLAST. There was very little biological interpretation. However, the next step of this research would be to interpret the output more intensely at a biological level and also relate them to real life problems, mainly targeted to the people of Bangladesh.

3.2. Purpose of the Research:

The number of cancer patients is increasing day by the day in Bangladesh. The methods of diagnosis are often slow and the treatment is stressful for patients in most cases. Currently, the only treatment available in the country is chemotherapy and radiotherapy which takes a heavy toll on the patient's health. At the same time the treatments are very expensive and do not guarantee total recovery for the patient.

There is also another area where this research can be applied. That is, treating mental patients. It is often believed that mental illnesses are hereditary. This implies that mental illnesses must have a genetic component associated with it. Bangladesh does not have a very sophisticated treatment system for mental patients. This area could be tapped by bioinformatics for the treatment of mental patients.

The future plan of this research is to focus on these two areas at a more biological level. The main aim would be to find genetic information about cancer and mental patients and see if genetic treatment can be done to ease the treatment of these patients, both in terms of their health and cost.

4.0 CONCLUSION:

Working with bioinformatics has proven to be a new and challenging task. A lot of exposure to DNA analysis is still needed to come up with something really meaningful. However, this paper has at least shown that BLAST can be used effectively in a country like Bangladesh for sequence alignment and information derivation. If proper exposure is found along with efficient use of BLAST, this can usher in a whole new era in the medical treatment and diagnosis process of Bangladesh.

5.0. REFERENCE:

Books:

[1] H.F. Jordan and G. Alaghband, *Fundamentals of Parallel Processing,* Prentice Hall of India, 2002.

[2] J. Bedell, I. Korf and M. Yandell, *BLAST*, O' Reilly, 2003.

[3] Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997.

[4] Lesk, *Introduction to Bioinformatics*, Oxford University Press, 2005.

Articles:

[1] T. Madden, *The BLAST Sequence Analysis Tool*, 2003.

Lectures:

[1] S. Brown, *Overview and Introduction to Database Searching and Pairwise Alignments*, NYU School of Medicine, 2002.

URLs:

[1] http://www.ncbi.nlm.nih.gov/blast/BLAST_guide.pdf

[2] http://www.scq.ubc.ca/?p=385

[3] http://bioinformatics.oxfordjournals.org/current.dtl

APPENDIX -1:

Complete output of a sequence:

Here, a nucleotide sequence was entered and run against an environmental database.

```
BLASTN 2.2.14 [May-07-2006]


Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A.
Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database
search
programs",  Nucleic Acids Res. 25:3389-3402.
Database: environmental samples
           1,545,740 sequences; 1,212,001,233 total letters

Searching

Query= Test
         (210 letters)




                                                                 Score
E
Sequences producing significant alignments:                      (bits)
Value

gb|AACY01088767.1| Environmental sequence IBEA_CTG_1979754, whol...
123   2e-26
gb|AATN01004650.1| Environmental sequence ctg3353, whole genome ...
38   0.74
gb|AACY01044310.1| Environmental sequence IBEA_CTG_1977624, whol...
38   0.74
gb|AACY01331945.1| Environmental sequence IBEA_CTG_SSBOF36TF, wh...
38   0.74
gb|AACY01368201.1| Environmental sequence IBEA_CTG_SVADQ07TR, wh...
38   0.74
gb|AACY01515664.1| Environmental sequence IBEA_CTG_UAAVU86TR, wh...
36   2.9
gb|AAFX01074022.1| Environmental sequence XZS116217.b1, whole ge...
36   2.9
```

```
gb|AACY01162823.1| Environmental sequence IBEA_CTG_SHAAB38TF, wh...
36   2.9
gb|AACY01027370.1| Environmental sequence IBEA_CTG_2154041, whol...
36   2.9
gb|AACY01052399.1| Environmental sequence IBEA_CTG_2000097, whol...
36   2.9
gb|AACY01057602.1| Environmental sequence IBEA_CTG_2157890, whol...
36   2.9
gb|AACY01089073.1| Environmental sequence IBEA_CTG_2042967, whol...
36   2.9
gb|AACY01100082.1| Environmental sequence IBEA_CTG_1958456, whol...
36   2.9
gb|AACY01118950.1| Environmental sequence IBEA_CTG_2094034, whol...
36   2.9
gb|AACY01137643.1| Environmental sequence IBEA_CTG_2036568, whol...
36   2.9
gb|AACY01170227.1| Environmental sequence IBEA_CTG_SKAC622TR, wh...
36   2.9
gb|AACY01287238.1| Environmental sequence IBEA_CTG_SSAZ164TF, wh...
36   2.9
gb|AACY01337740.1| Environmental sequence IBEA_CTG_SSBQ719TR, wh...
36   2.9
gb|AACY01376731.1| Environmental sequence IBEA_CTG_SXAC655TF, wh...
36   2.9
gb|AACY01382148.1| Environmental sequence IBEA_CTG_SXAF902TR, wh...
36   2.9
gb|AACY01435076.1| Environmental sequence IBEA_CTG_SZAQU83TR, wh...
36   2.9
gb|AACY01485340.1| Environmental sequence IBEA_CTG_UAALH03TF, wh...
36   2.9

>gb|AACY01088767.1| Environmental sequence IBEA_CTG_1979754, whole
genome shotgun
           sequence
          Length = 1172

 Score =  123 bits (62), Expect = 2e-26
 Identities = 74/78 (94%)
 Strand = Plus / Plus


Query: 128 atactttaaccaatataggcatagcgcacagacagataaaaattacagagtacacaacat
187
           ||||||||||||||||||||| ||  ||||||||||||||||||||||||||||||||||
Sbjct: 794 atactttaaccaatataggcacaggacacagacagataaaaattacagagtacacaacat
853


Query: 188 ccatgaaacgcattagca 205
           ||||||||||||| ||||
Sbjct: 854 ccatgaaacgcatcagca 871
```

```
>gb|AATN01004650.1| Environmental sequence ctg3353, whole genome
shotgun sequence
          Length = 1364

 Score = 38.2 bits (19), Expect = 0.74
 Identities = 19/19 (100%)
 Strand = Plus / Plus



Query: 98  attaaaattttattgactt 116
            |||||||||||||||||||
Sbjct: 308 attaaaattttattgactt 326


>gb|AACY01044310.1| Environmental sequence IBEA_CTG_1977624, whole
genome shotgun
           sequence
          Length = 1739

 Score = 38.2 bits (19), Expect = 0.74
 Identities = 19/19 (100%)
 Strand = Plus / Minus


Query: 95  taaattaaaattttattga 113
            |||||||||||||||||||
Sbjct: 878 taaattaaaattttattga 860


>gb|AACY01331945.1| Environmental sequence IBEA_CTG_SSBOF36TF, whole
genome shotgun
           sequence
          Length = 919

 Score = 38.2 bits (19), Expect = 0.74
 Identities = 19/19 (100%)
 Strand = Plus / Plus


Query: 98  attaaaattttattgactt 116
            |||||||||||||||||||
Sbjct: 878 attaaaattttattgactt 896


>gb|AACY01368201.1| Environmental sequence IBEA_CTG_SVADQ07TR, whole
genome shotgun
           sequence
          Length = 861
 Score = 38.2 bits (19), Expect = 0.74
 Identities = 19/19 (100%)
 Strand = Plus / Minus


Query: 99  ttaaaatttattgactta 117
           |||||||||||||||||||
Sbjct: 388 ttaaaatttattgactta 370


>gb|AACY01515664.1| Environmental sequence IBEA_CTG_UAAVU86TR, whole
genome shotgun
           sequence
          Length = 567

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Plus


Query: 103 aattttattgacttaggt 120
           ||||||||||||||||||
Sbjct: 287 aattttattgacttaggt 304


>gb|AAFX01074022.1| Environmental sequence XZS116217.b1, whole genome
shotgun sequence
          Length = 1035

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Minus


Query: 180 cacaacatccatgaaacg 197
           ||||||||||||||||||
Sbjct: 600 cacaacatccatgaaacg 583


>gb|AACY01162823.1| Environmental sequence IBEA_CTG_SHAAB38TF, whole
genome shotgun
           sequence
          Length = 1002

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Minus


Query: 93  agtaaattaaaattttat 110
           ||||||||||||||||||
Sbjct: 829 agtaaattaaaattttat 812
```

```
>gb|AACY01027370.1| Environmental sequence IBEA_CTG_2154041, whole
genome shotgun
           sequence
          Length = 2642


 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Minus



Query: 98  attaaaatttattgact 115
            ||||||||||||||||||
Sbjct: 562 attaaaatttattgact 545


>gb|AACY01052399.1| Environmental sequence IBEA_CTG_2000097, whole
genome shotgun
           sequence
          Length = 1145

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Plus




Query: 94  gtaaattaaaatttatt 111
            ||||||||||||||||||
Sbjct: 206 gtaaattaaaatttatt 223


>gb|AACY01057602.1| Environmental sequence IBEA_CTG_2157890, whole
genome shotgun
            sequence
          Length = 3892

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Minus


Query: 99   ttaaaatttattgactt 116
             ||||||||||||||||||
Sbjct: 1580 ttaaaatttattgactt 1563


>gb|AACY01089073.1| Environmental sequence IBEA_CTG_2042967, whole
genome shotgun
            sequence
```

```
      Length = 1786

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Plus




Query: 67   agcttctgaactggttac 84
            ||||||||||||||||||
Sbjct: 1629 agcttctgaactggttac 1646


>gb|AACY01100082.1| Environmental sequence IBEA_CTG_1958456, whole
genome shotgun
           sequence
          Length = 1794

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Plus


Query: 95  taaattaaaattttattg 112
           ||||||||||||||||||
Sbjct: 252 taaattaaaattttattg 269


>gb|AACY01118950.1| Environmental sequence IBEA_CTG_2094034, whole
genome shotgun
            sequence
          Length = 1952

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Minus


Query: 125  aaaatactttaaccaata 142
            ||||||||||||||||||
Sbjct: 1124 aaaatactttaaccaata 1107


>gb|AACY01137643.1| Environmental sequence IBEA_CTG_2036568, whole
genome shotgun
           sequence
          Length = 944

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 21/22 (95%)
 Strand = Plus / Plus
```

```
Query: 96  aaattaaaattttattgactta 117
           |||||| |||||||||||||||
Sbjct: 546 aaattataattttattgactta 567


>gb|AACY01170227.1| Environmental sequence IBEA_CTG_SKAC622TR, whole
genome shotgun
           sequence
          Length = 902

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Plus


Query: 96  aaattaaaattttattga 113
           ||||||||||||||||||
Sbjct: 386 aaattaaaattttattga 403


>gb|AACY01287238.1| Environmental sequence IBEA_CTG_SSAZ164TF, whole
genome shotgun
           sequence
          Length = 894

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Minus


Query: 99  ttaaaattttattgactt 116
           ||||||||||||||||||
Sbjct: 524 ttaaaattttattgactt 507


>gb|AACY01337740.1| Environmental sequence IBEA_CTG_SSBQ719TR, whole
genome shotgun
           sequence
          Length = 920

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 21/22 (95%)
 Strand = Plus / Minus


Query: 96  aaattaaaattttattgactta 117
           |||||| |||||||||||||||
Sbjct: 109 aaattataattttattgactta 88


>gb|AACY01376731.1| Environmental sequence IBEA_CTG_SXAC655TF, whole
genome shotgun
           sequence
```

```
            Length = 929

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Plus


Query: 98  attaaaattttattgact 115
           ||||||||||||||||||
Sbjct: 300 attaaaattttattgact 317


>gb|AACY01382148.1| Environmental sequence IBEA_CTG_SXAF902TR, whole
genome shotgun
           sequence
          Length = 909

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Plus


Query: 67  agcttctgaactggttac 84
           ||||||||||||||||||
Sbjct: 612 agcttctgaactggttac 629


>gb|AACY01435076.1| Environmental sequence IBEA_CTG_SZAQU83TR, whole
genome shotgun
           sequence
          Length = 418

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 21/22 (95%)
 Strand = Plus / Minus


Query: 100 taaaattttattgacttaggtc 121
           |||||||||||| |||||||||
Sbjct: 308 taaaattttattcacttaggtc 287


>gb|AACY01485340.1| Environmental sequence IBEA_CTG_UAALH03TF, whole
genome shotgun
           sequence
          Length = 752

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 18/18 (100%)
 Strand = Plus / Minus
```

```
Query: 93  agtaaattaaaattttat 110
           ||||||||||||||||||
Sbjct: 86  agtaaattaaaattttat 69


  Database: environmental samples
    Posted date:  Oct 31, 2006 10:16 AM
  Number of letters in database: 1,212,001,233
  Number of sequences in database:  1,545,740

Lambda     K      H
    1.37    0.711     1.31

Gapped

Lambda     K      H
    1.37    0.711     1.31


Matrix: blastn matrix:1 -3
Gap Penalties: Existence: 5, Extension: 2
Number of Sequences: 1545740
Number of Hits to DB: 3,298,515
Number of extensions: 222207
Number of successful extensions: 65701
Number of sequences better than 10.0: 22
Number of HSP's gapped: 65701
Number of HSP's successfully gapped: 22
Length of query: 210
Length of database: 1,212,001,233
Length adjustment: 19
Effective length of query: 191
Effective length of database: 1,182,632,173
Effective search space: 225882745043
Effective search space used: 225882745043
X1: 11 (21.8 bits)
X2: 15 (29.7 bits)
X3: 25 (49.6 bits)
S1: 12 (24.3 bits)
S2: 18 (36.2 bits)
```

APPENDIX -2:

Interpretation of the result:

It has been found that the nucleotide sequence matched with the database in most cases. However, the matching might be considered to be a lucky one in this case as most sequences produced a very high E- value. This means that in most cases the sequences matched by chance. The results could have been different if a different database was chosen.

One other very significant aspect of the output is the fact that some sequences might never produce a protein when they interact with each other. One such example is given below:

```
>gb|AACY01435076.1| Environmental sequence IBEA_CTG_SZAQU83TR, whole
genome shotgun
           sequence
          Length = 418

 Score = 36.2 bits (18), Expect = 2.9
 Identities = 21/22 (95%)
 Strand = Plus / Minus

Query: 100 taaaattttattgacttaggtc 121
           |||||||||||| |||||||||
Sbjct: 308 taaaattttattcacttaggtc 287
```

Here, the strand output of Plus/Minus shows that the two sequences match in reverse order with respect to the database. It can be seen that the query sequence starts from 100 to 121 while the database sequence starts from 308, moves in the reverse direction and ends at 287 in order to align the sequence. These evidence prove that the production of a new protein sequence is not possible through the interaction of these sequences.

APPENDIX -3:

Installing BLAST: How TO:

Introduction:

BLAST stands for Basic Local Alignment Search Tool. It was first developed by the National Centre for Bioinformatics Information (NCBI) and used to search protein and nucleotide sequences to match local similarity between sequences. It also provides a statistical analysis of the matches. It may be mentioned that the BLAST software is available for free download at the following location:

http://www.ncbi.nlm.nih.gov/BLAST/

How to Install:

It should be noted that BLAST can only work in a Linux environment and hence the machine used to run BLAST must have Linux installed in it.

Once downloaded, the BLAST software would be in the form of a tar file which has to be unpacked. Typically, the name of the file would be ncbi.tar.gz and can be unpacked using the following command:

```
$ tar –zxf ncbi.tar.gz
```

Building BLAST:

After the tar file is properly unpacked, a separate folder named **ncbi** would be created containing the following files:

1. ncbi/doc/FAQ.txt
2. ncbi/make/readme.unx

These two files give instructions on how to build the software. In fact, we execute the following command to build it:

```
$ ./ncbi/make/makedis.csh >& make.log
```

It might take a while to finish building the software. When the building process is complete, the log messages are stored in **make.log** and the binaries are stored in **./ncbi/build** directories as well as the **./ncbi/bin** directory.

Use of Database and Environmental variables:

The databases against which the sequences are to be matched have to be downloaded before the BLAST software could be used. The databases can be downloaded from the following location:

ftp://ftp.ncbi.nlm.nih/gov/blast/db/

The downloaded databases are preformatted and do not need any further formatting process. However, the databases are saved in the form of tar files which have to unpacked before they could be put to use. A simple example of unpacking a database is given below:

Name of database: env_nt.tar.gz

Command used: $ tar –zxf env_nt.tar.gz

In the meantime we need to define two environmental variables before we can use BLAST properly. The variables are:

- **$BLASTDB:** This is the variable that will point to the BLAST database. If the software is installed in the home folder of the user, then $BLASTDB could be set to the following location:

  $HOME/ncbi_blast/ncbi/db/

- **$BLASTMAT:** This is the variable that points to the scoring matrix. It is set to the following location:

  $HOME/ncbi_blast/ncbi/data/

Creating an Input Sequence:

BLAST requires two sets of datasets. One is the input sequence while the other is the database against which the query is done. The test input file could be created in the following manner from the Linux terminal:

```
cat > test.txt
    >Test

    AGCTTTTCATTCTGAACGTATAGTACAAAAAGAGTGTGAGCAGCTTCTGA
    ACTGGTGCACCTTAAATTTTATTGACTTAGGTCAAAATAACCATATAACCA
    CCCTAGTACTTTGACTCCCCCCTATAGCGGGTAAAGCGCTCCCTAGTAT
    A.
```

The above command will create a **.txt** file named **test** and store the sequence given above that will be used as the input sequence.

Running BLAST using a Shell file:

Using a shell file makes life a lot easier to run BLAST. The shell file script is given below:

```
# ! /bin/bash
#
#$ -cwd
#$ -S /bin/bash
#$ -j y

export BLASTDB=$HOME/ncbi-blast/ncbi/db/
export BLASTMAT=$HOME/ncbi-blast/ncbi/data/

export PATH=$PATH:$HOME/ncbi-blast/ncbi/bin

blastall -d env_nt -p blastn -i $HOME/test.txt -o $HOME/result.txt
```

The script is written assuming that the user has stored the BLAST software in his home folder. In case of a different location, the path of the environmental variables has to be set to the appropriate location.

The script initially sets the environmental variables to point to the database and the scoring matrices. Then the path to the binaries is set using the PATH variable. The last line denotes the actual running of the program. It runs the **blastall** program searching the **env_nt** database and looks for a **nucleotide** (denoted by **blastn**) sequence match. The input sequence is gathered from the **test.txt** file and the output is saved in the **result.txt** file.

This shell file is then saved at an appropriate location. In order to run the file, the following command is used:

```
chmod a+x blast_sge.sh
. blast_sge.sh
```

Once the above commands are executed, the output is stored in the **results.txt** file as mentioned above.

THE END