Design Style of BLAST and FASTA and Their Importance in Human Genome.

Saba Khalid[1] and Najam-ul-haq[2]
SZABIST
Karachi, Pakistan

Abstract: *This subjected study will discuss the concept of BLAST and FASTA.BLAST are used to compare biological sequences against a public database and similarly also used for pattern matching of any unknown gene or sequences, whereas FASTA algorithm is basically an improved tool that introduces for the DNA searches i.e. it added the facility to do DNA sequence searches. Lastly, I have prepare a comparative study that focuses on pros and cons of both BLAST and FASTA, newly features that are added to both BLAST and FASTA after they being came into context and which algorithm is most widely used in the field of biological era.*

Keywords: *BLAST, FASTA, DNA and human genome.*

1. INTRODUCTION

Over the past 15 years, an era has come where searching for any query sequence become time consuming whether it is a protein sequence, a DNA sequence or a nucleotide sequence. As sequence contains large no of amino acids or proteins for which comparing them is quite difficult. Alignment of large sequence is also time taken tasks. Before BLAST and FASTA came into being, a well known algorithm called Smith-Waterman works for database searches for a protein sequence or any other sequence. Smith-Waterman algorithm is somewhat similar to BLAST and FASTA algorithm but it is too slow to perform database searches for a large query sequences because it uses a full alignment procedure, which result in wastage of time and computer power and intensity.[10]

In 1988, FASTA algorithm came into being." FASTA" means fast alignment. FASTA algorithm came into context in 1988 and it was developed by Pearson and Lipman. FASTA algorithm is the first fast sequence searching algorithm for comparing a user query against an available database. FASTA algorithm is basically an improved tool that introduces for the DNA searches i.e. it added the facility to do DNA sequence searches.

BLAST (Basic Local Alignment Search Tool) algorithm came in context in 1990; this is first BLAST algorithm which was developed by Steve Altschul, Warren Gish and Dave Lipman in 1990 at the National Center for Biotechnology Information (NCBI). BLAST are used to compare biological sequences against a public database and similarly also used for pattern matching of any unknown gene or sequences. BLAST algorithm and programs has been typically designed for speed. Hence BLAST algorithm will be extremely useful in knowing whether user query sequence is related to any other genome or proteins. BLAST algorithm mainly focuses on speed so that this makes BLAST algorithm more practical towards large sequences against huge genome databases.

2. BLAST

BLAST stands for Basic Local Alignment Search Tool. Basically BLAST is a typical algorithm for comparing any sequences e.g. any biological sequences with other sequences whether it is a nucleotide sequence, protein sequence or DNA sequence.

BLAST typically is used to perform sequences in biological manner against a public database and similarly also used for pattern matching of any unknown gene or sequences. BLAST algorithm and programs has been typically designed for speed. BLAST algorithm doesn't comprise on speed and on the other hand it has a minimal or least sacrifice on sensitivity in a distant relationship of sequences. BLAST program uses a heuristic algorithm that finds local as well as global alignments this is why it is able to find relationships between sequences that have some extent of similarity. [1]

BLAST was developed by Stephen Altschul, Warren Gish, and David Lipman in 1990. NCBI(National Center for Biotechnology Information)is the website where the BLAST algorithm is access able and also NCBI is the website where we can find the different gene to be tested and also different patterns to be searched e.g. human, rat, mouse etc.

2.1 BLAST Diagram

Following diagram shows how BLAST algorithm takes input from user perspective

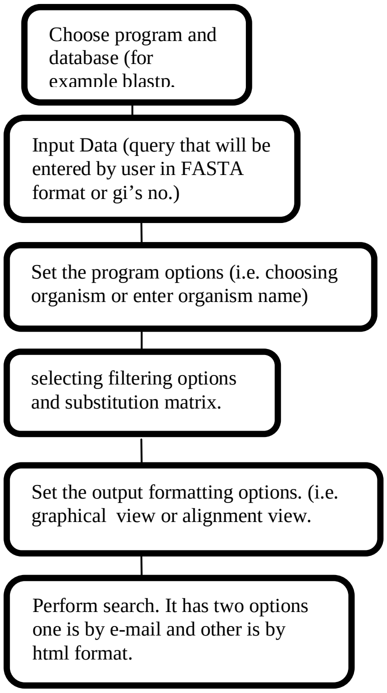

Choose program and database (for example blastn.
Input Data (query that will be entered by user in FASTA format or gi's no.)
Set the program options (i.e. choosing organism or enter organism name)
selecting filtering options and substitution matrix.
Set the output formatting options. (i.e. graphical view or alignment view.
Perform search. It has two options one is by e-mail and other is by html format.

Figure 1: basic BLAST algorithm flow chart

2.2 BLAST Program

The BLAST algorithm allows us to select a program according to criteria of search. There are different BLAST program options available some most commonly used programs are listed below.

BLAST PROGRAM	PURPOSE
nucleotide blast	Compares and Uses a nucleotide query with respect to the nucleotide database i.e.(blastn)
protein blast	Compares and Uses a protein query with respect to the protein database i.e.(blastp, psi-blast)
Gene blast	Compares and uses a human genome sequences with mouse genome sequences .i.e.(blastz)
Mega blast	It searches for similar DNA sequences that are highly associated with each other in rapid speed.
blastx	Using a **translated nucleotide** query for the search of protein database.
WU blast	Enhanced version of blast that uses gapped alignments.
tblastn	Compares and uses a **protein** query with respect to the search of translated nucleotide.

2.3 BLAST Input

In BLAST algorithm, there are different ways to give inputs according to any particular user need or requirements. BLAST inputs satisfy user need by allowing three types of inputs. These three inputs are defined below:

2.3.1 FASTA Format

In FASTA format sequence has a greater-than (">") sign or symbol which is the main distinguish between other formats and FASTA formats. The FASTA format has sequence that begins with single-line description of the whole sequence. The whole started with symbol (">") and a line description followed by a sequence.
For example:

```
gi|1345176|sp|P01113|OVAX_CHICK GENE X PROTEIN
(OVALBUMIN-RELATED)

QIKDLLNCHTEWJXCJSLVLVNAIYFKGMWKTAFNAEDTRE
MPFHVTKQESKPVQMMCEGNTYQNXMKEAE
KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEW
TNPNTMEKRRVKVYLPQMKIEEKYNLTS
VLMALGMTERCFWITHNYRSSAESLKISQAVHGAFMELSED
GIFDFMAGSTGVSDFKJDHFJDHFJSDFHJHCJDHFDJHFJDHJ
JFHJD
```

Here, I have taken a typical OVAX_CHICK GENE X PROTEIN and performed all inputs using this protein.
The above example shows that there is a how a FASTA format sequence can be identified. The most important point to be noted is that there should be no blank in whole FASTA sequence format.
FASTA format sequences should always be represented in IUB/IUPAC i.e. amino acids and nucleic acid codes. There are also some exceptions which should also be considered for example I) lower-case letters will be converted in upper-case letters by mean of mapping. ii) A dash or hyphen can represent a gap in a sequence length.

2.3.2 Bare Sequence Format

Bare sequence format represent a lines of sequences with no starting line description. Bare sequence format example is as follows:

```
QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTR
EMPFHVTKQESKPVQMMCMNNSFNVATLPAE
KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTE
WTNPNTMEKRRVKVYLPQMKIEEKYNLTS
VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSE
DGIEMAGSTGVIEDIKHSPESEQFRADHP
FLFLIKHNPTNTIVYFGRYWSP
```

2.3.3 Identifiers

Identifiers are actually accession number or accession version or gi's. For example (p98213, ASAA66326 or 165789).Identifiers can also be recognized by bar separated identifiers which is also known as NCBI sequence identifier. For example (gi|172345).in identifier, spaces can be put on before or after the identifier will declared otherwise it be treated as bare sequence format. Example of incorrect deceleration of identifier is (gi| 172345).

2.4 BLAST Output

BLAST Output can be understood according to user need. BLAST Output is accessible and can be viewed in several different ways by which a user can understand and satisfy its entered query.

There are three different ways by which a BLAST Output can be viewed and understand. These ways are defined below:

2.4.1 Graphical Display

The first and basic type of BLAST output is graphical display, which is most easy human understandable output for any user to view and understands it query output. Graphical Display output defines how much portion of another sequence matches to your query sequences. Following example diagram shows the output of a basic graphical display of any query entered in BLAST algorithm.

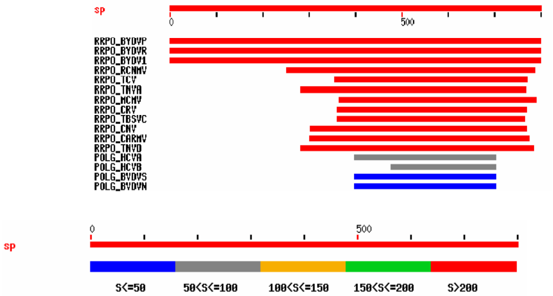
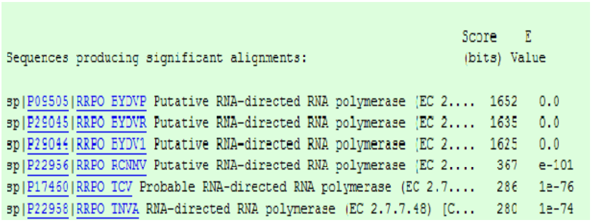


Figure 2: graphical display of BLAST output

A red, green and yellow match shows good or significant matches in both queries. Similarly, grey matches shows intermediate matches and lastly blue shows bad matches i.e. less matches in both query.

2.4.2 HIT List

The second type of BLAST Output is hit list. Hit list basically provides the name of sequences that is similar to your (user) sequences. These names of sequences will be ranked by similarity basics. Following example diagram shows the output of hit list display of any query entered in BLAST algorithm.



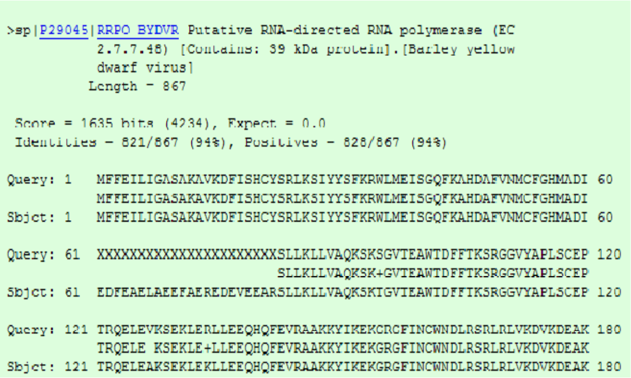
```
                                                                Score    E
Sequences producing significant alignments:                     (bits) Value

sp|P09505|RRPO BYDVP Putative RNA-directed RNA polymerase (EC 2....  1652   0.0
sp|P29045|RRPO BYDVR Putative RNA-directed RNA polymerase (EC 2....  1635   0.0
sp|P29044|RRPO BYDV1 Putative RNA-directed RNA polymerase (EC 2....  1625   0.0
sp|P22956|RRPO RCNMV Putative RNA-directed RNA polymerase (EC 2....   367   e-101
sp|P17460|RRPO TCV Probable RNA-directed RNA polymerase (EC 2.7....   286   1e-76
sp|P22958|RRPO TNVA RNA-directed RNA polymerase (EC 2.7.7.48) [C...   280   1e-74
```

Figure 3: hit list display of BLAST output

- **Sequence no and names** are exactly what the database entry for BLAST hit list algorithm.
- **Description** defines particular sequences definition or defining the sequence.
- **Score or bit value** defines how much percent similarity is there between two sequences. Basically bit value is the measurement of similarity between two sequences. The higher the bit values the better matches between sequences.
- **E-value** Those sequences identical to the query must have any E-value 0.normally and standard proves that if a user need a certain type of homology sequences it must have E-value lower than 10^4.

2.4.3 Alignment

The third type of BLAST Output is alignment. The output of this type shows every alignment in user query followed by percentage of identical alignments in user query. Following example diagram shows the output of alignment display of user query entered in BLAST algorithm.



```
>sp|P29045|RRPO BYDVR Putative RNA-directed RNA polymerase (EC
          2.7.7.48) [Contains: 39 kDa protein].[Barley yellow
          dwarf virus]
         Length = 867

 Score = 1635 bits (4234), Expect = 0.0
 Identities = 821/867 (94%), Positives = 828/867 (94%)

Query: 1   MFFEILIGASAKAVKDFISHCYSRLKSIYYSFKRWLMEISGQFKAHDAFVNMCFGHMADI 60
            MFFEILIGASAKAVKDFISHCYSRLKSIYYSFKRWLMEISGQFKAHDAFVNMCFGHMADI
Sbjct: 1   MFFEILIGASAKAVKDFISHCYSRLKSIYYSFKRWLMEISGQFKAHDAFVNMCFGHMADI 60

Query: 61  XXXXXXXXXXXXXXXXXXXXXSLLKLLVAQKSKSGVTEAWTDFFTKSRGGVYAPLSCEP 120
                                 SLLKLLVAQKSK+GVTEAWTDFFTKSRGGVYAPLSCEP
Sbjct: 61  EDFEAELAEEFAEREDEVEEARSLLKLLVAQKSKTGVTEAWTDFFTKSRGGVYAPLSCEP 120

Query: 121 TRQELEVKSEKLERLLEEQHQFEVRAAKKYIKEKGRGFINCWNDLRSRLRLVKDVKDEAK 180
            TRQELE KSEKLE+LLEEQHQFEVRAAKKYIKEKGRGFINCWNDLRSRLRLVKDVKDEAK
Sbjct: 121 TRQELEAKSEKLEKLLEEQHQFEVRAAKKYIKEKGRGFINCWNDLRSRLRLVKDVKDEAK 180
```

Figure 4: Alignment display of BLAST output

- A good alignment should not have too many gaps in sequences and also should have less complexity regions for a good calculated percentage of identical sequences.

2.5 Basic Working of BLAST Algorithm

BLAST algorithm works in heuristic manner, basically a typical BLAST algorithm uses a heuristic approach to find the similarity in a query sequence and also other features according to user need.

To start work in BLAST algorithm first of all we need a query sequence to run on BLAST. And secondly need another sequence for which we are looking for, that will search against first sequence to find out how much

similarity both sequence contains. BLAST will take out those subsequences from the databases which are familiar to those subsequences in your (user) query.
The important point to be noted is that query sequence enter by user must be smaller than the query present in the database. BLAST algorithm is 50 times faster than traditional Smith-Waterman algorithm.
Following are the steps for a typical BLAST protein (balstp) algorithm.[1]

- Firstly, we have to remove those regions which are of low-complexity or the subsequences that repeats itself again and again in a query sequence. These regions might contain high scoring points which will be marked by X for a protein sequence.

- Second step is most important; in a typical BLAST now it will break the query into words. In protein sequence it will take k=3 means length of three words in a typical protein sequence. Following diagram will show the iteration of the words in a query sequence, that is how a word list will be formed in a typical protein sequence.

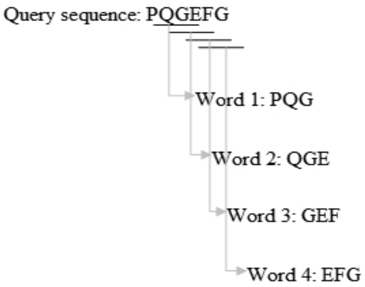


Figure: 5 how a word list is formed in protein sequence.

BLAST algorithm will now see the high scoring words in the protein sequences i.e. the scores will be created by comparing the set of words with all 3-letter word. The matches will score a +5 value and a mismatch will score a -4 value in a scoring matrix. The scoring matrix used is substitution matrixes that contain all the value of matches or mismatches of words in a protein sequence.

After scoring the BLAST program will quickly perform all the words that leads to the high scoring words and then compare these words to the database sequences i.e. it will identify all the exact matches with the available database sequences.

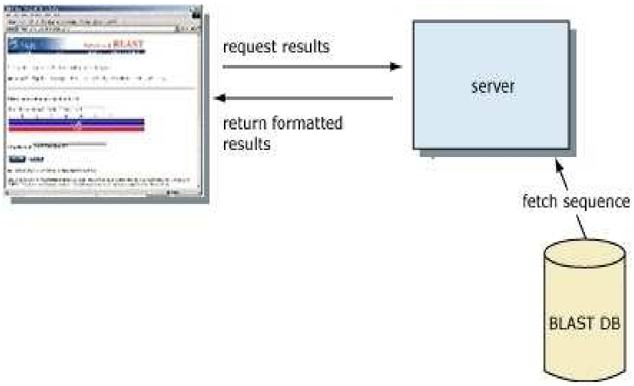


Figure 6: identifying all the exact matches with the available database sequences.

- Once exact matches with databases is done BLAST algorithm will figure out how good alignment is being done to have a possible and authentic biological relationship. The bit score and E-value i.e. expect value is produced by the BLAST program to observe the similarity between the sequences.

- Bit score tells the user how good alignment is done between sequences. Basically it is identification of finding out the rate of alignment in both sequences. The key element for finding out how good alignment is done between two sequences is a substitution matrix. The BLOSUM62 matrix is by default use in protein sequences and by mostly BLAST programs.

- Those sequences identical to the query must have any E-value 0.normally and standard proves that if a user need a certain type of homology sequences it must have E-value lower than 10^4.The lower the value the good, better and more significant the hits are done.

- The above all the steps is significant and important in performing a typical BLAST program for a protein sequence and lastly these steps also varies if a BLAST program uses different type of sequences i.e. other than protein sequence for example DNA sequence or any other.

2.6 Example of BLAST Program

A very good and significant example is being performed using BLAST to show how it works for human genome and other biological species and relations.
BLAST will examine and taken the protein coat (capsid), a well-known and one of the most dangerous virus. This is called West Nile Virus. This virus can infect human, animals like horses and birds. This virus is transmitted by mosquitoes which infect the human and animals badly.
A ribbon diagram below shows a 3D protein structure of the infected virus capsid.

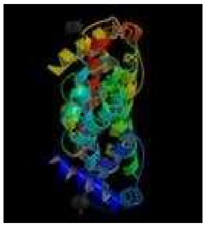


Figure 7: 3D protein structure of capsid virus

The biological sequence of this protein is:

RVLSLTGLKRAMLSLIDGRGPTRFVLALLAFFRFTAIAPT RAVLDRWRSVNKQTAMKHLLSFKKELGTLTSAINRR

As the BLAST program signify separate and individual amino acids in this above protein sequence. Now there is another sequence which is targeted sequence against the capsid virus protein sequence. This target sequence is revealed from NCBI database.

RVLSLTGLKRAMLSLIDGRGPTRFVLALLAFFRFTAIAPT RAVLDRWRSVNKQTAMKHLL

Now the above sequence is linked or associated with Kunjin Virus. Now by placing both sequences side by side with each other it is revealed that this Kunjin Virus is up closely related to West Nile Virus.

RVLSLTGLKRAMLSLIDGRGPTRFVLALLAFFR FTAIAPTRAVLDRWRSVNKQTAMKHLLSFKKE LGTLTSAINRR

RVLSLTGLKRAMLSLIDGRGPTRFVLALLAFFR FTAIAPTRAVLDRWRSVNKQTAMKHLL

By placing both sequences with each other it clearly shows similarity of both sequences. Surprisingly, it's true that either sequences or in other words both species have extendable kind of similarities in their protein coats.

BLAST program and algorithm correctly identifies the relationships among sequences, proteins and biological relationships. This could provide a more useful and authentic future studies on the nature of viral protein structure, their similarities and what measure should be taken in a correct way to help preventing human lives from these viruses by taking right king of vaccines against these dangerous virus.

3. FASTA

In 1988, FASTA algorithm came into being." FASTA" means fast alignment. FASTA algorithm came into context in 1988 and it was developed by Pearson and Lipman. FASTA algorithm is the first fast sequence searching algorithm for comparing a user query against an available database.

FASTA algorithm is basically an improved tool that introduces for the DNA searches i.e. it added the facility to do DNA sequence searches. FASTA works in a dot plot manner; it takes an amino-acid sequences or any other sequences and searches for other corresponding sequence by using local alignment of sequences so that it can find out the matches of similarity in database sequences. [9]

FASTA algorithm initial version has somewhat been discarded and an online toolbox is now used if a user need to run a query on FASTA algorithm. This online toolbox is called "Fa-box" which perform user task related to FASTA algorithm.

3.1 Working of FASTA Algorithm

Following few diagrams shows how FASTA algorithm works and in which manner similarity search sequence task takes place.

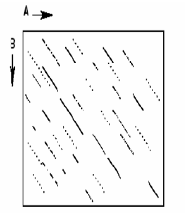


Figure 8[9]: localizing the similarity search regions.

In figure 8, FASTA algorithm is localizing the similarity search regions between the two sequences i.e. the user query sequence and target sequence in database. Each identity between the sequences is represented by dark dash line.

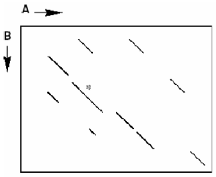


Figure 9[9]: score the ten best similarity search.

In figure 8, the second step is to score the ten best score words in both sequences using a scoring matrix.

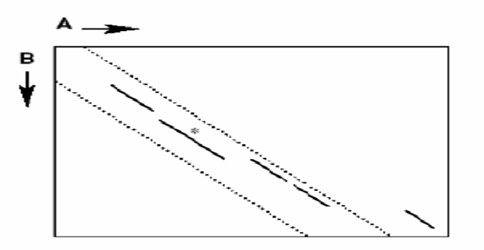


Figure 10: finding way for best combination.

In figure 9, the third step is to to find a way that fit to diagonal shape so that best combination of diagonal score came into context. In this a diagonal includes highest scoring segments

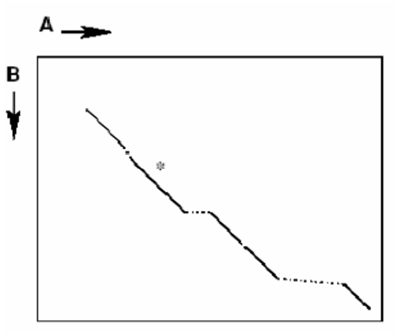


Figure 11: joining the segments.

In figure 10, segments that had been selected by best score are now joined by performing dynamic programming so that an optimal alignment can be created.

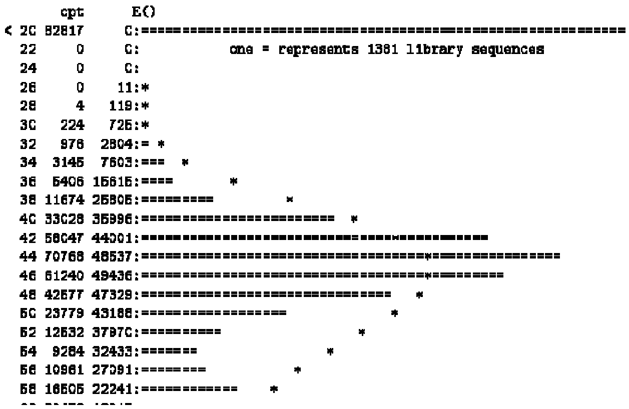
3.2 FASTA Output

FASTA output can view and understood according to user need. FASTA Output is accessible and can be viewed in several different ways by which a user can understand and satisfy its entered query. There are different ways by which a FASTA Output can be viewed and understand. These ways are defined below:

3.2.1 Histogram in FASTA Output

Following diagram shows FASTA output known as Histogram. The X axis is the score, printed on the left column. The y axis shows the number of matching database records having the score. The expected random distribution of the score is shown by ``*" signs. For example, a score of 34 was attained by 1045 sequences when the query was searched, compared to 1564 expected sequences with a random sequence search.

Each Line describes one database sequence matching the query, printed in decreasing order of statistical significance. Each contains the name of the record, its database ID, a short description of the sequence.

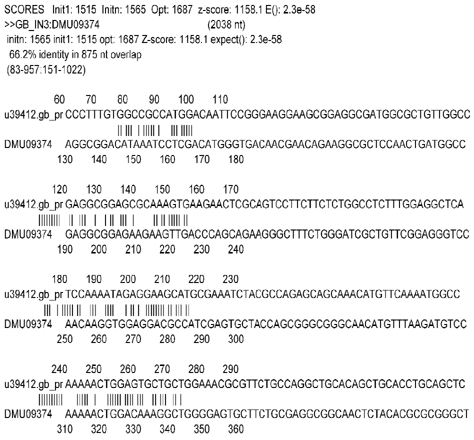


```
       opt      E()
< 20 82817     0:==========================================================
  22     0     0:           one = represents 1381 library sequences
  24     0     0:
  26     0    11:*
  28     4   119:*
  30   224   725:*
  32   976  2804:= *
  34  3145  7603:===  *
  36  5406 15615:====       *
  38 11674 25805:=========         *
  40 33028 35996:========================  *
  42 58047 44001:===========================================
  44 70768 48537:=======================================*================
  46 61240 49436:=======================================*=========
  48 42577 47329:===============================   *
  50 23779 43188:==================            *
  52 12532 37970:==========                *
  54  9284 32433:=======               *
  56 10961 27091:========           *
  58 16505 22241:============    *
```

Figure 12: histogram output in FASTA format.

3.2.2 Alignment in FASTA Output

Following diagram shows the FASTA output in Alignment format. Alignment output shows that how both sequences are compared with each other so that percentage of identities and similarities can be shown in the form of output format.



```
SCORES  Init1: 1515  Initn: 1565  Opt: 1687  z-score: 1158.1 E(): 2.3e-58
>>GB_IN3:DMU09374                              (2038 nt)
 initn: 1565 init1: 1515 opt: 1687 Z-score: 1158.1 expect(): 2.3e-58
  66.2% identity in 875 nt overlap
 (83-957:151-1022)

              60        70        80        90        100       110
u39412.gb_pr CCCTTTGTGGCCGCCATGGACAATTCCGGGAAGGAAGCGGAGGCGATGGCGCTGTTGGCC
                          || ||| | ||||| |   ||| |||||
DMU09374     AGGCGGACATAAATCCTCGACATGGGTGACAACGAACAGAAGGCGCTCCAACTGATGGCC
               130       140       150       160       170       180

              120       130       140       150       160       170
u39412.gb_pr GAGGCGGAGCGCAAAGTGAAGAACTCGCAGTCCTTCTTCTCTGGCCTCTTTGGAGGCTCA
             |||||||  || |||  |  | || ||| |      || || ||||| ||
DMU09374     GAGGCGGAGAAGAAGTTGACCCAGCAGAAGGGCTTTCTGGGATCGCTGTTCGGAGGGTCC
               190       200       210       220       230       240

              180       190       200       210       220       230
u39412.gb_pr TCCAAAATAGAGGAAGCATGCGAAATCTACGCCAGAGCAGCAAACATGTTCAAAATGGCC
              ||| | ||||| ||   |||   ||||   | || |  ||||||| || ||| ||
DMU09374     AACAAGGTGGAGGACGCCATCGAGTGCTACCAGCGGGCGGGCAACATGTTTAAGATGTCC
               250       260       270       280       290       300

              240       250       260       270       280       290
u39412.gb_pr AAAAACTGGAGTGCTGCTGGAAACGCGTTCTGCCAGGCTGCACAGCTGCACCTGCAGCTC
             ||||||||   ||||| |   |||||| ||| |||  || |||  | |
DMU09374     AAAAACTGGACAAAGGCTGGGGAGTGCTTCTGCGAGGCGGCAACTCTACACGCGCGGGCT
               310       320       330       340       350       360
```

Figure: 13 Alignment output.

4. COMPARISON BETWEEN BLAST AND FASTA.

Lastly, I have studied whole design style of BLAST and FASTA and prepare a comparison study between BLAST and FASTA. This comparison study comprises of pros and cons of both BLAST and FASTA. The newly features that are added to both BLAST and FASTA after they being came into context. Secondly what new work is done by using BLAST and FASTA tools and techniques is being prepared in this comparison study. This comparison study has been prepared by review latest research papers.

The first comparison point between BLAST and FASTA is very basic i.e. what the enhancements are in case of version both algorithm and tools has been modified. By seeing this FASTA initial version is developed in 1988 and after that in 1998 FASTA lastly version is released called FASTA3.But on the other hand due to demanding power and ease of BLAST usage, BLAST variety of version came into context. PHI-BLAST, BLASTX, WU-BLAST these are some few version of BLAST that has different usage so that different requirement satisfying different user need. [4]

FASTA algorithm is the first fast sequence searching algorithm for comparing a user query against an available database. Whereas BLAST is concerned, BLAST (Basic Local Alignment Search Tool) is the improvement of FASTA algorithm. BLAST algorithm mainly focuses on speed so that this makes BLAST algorithm more practical towards large sequences against huge genome databases. And secondly BLAST algorithm is more easy to use as user gets a friendlier and readable output for its target entered query.

BLAST and FASTA also has a major difference in aspect of doing alignment. FASTA do only local alignment when calculating score for the similarity searches. While on the other hand BLAST do local as well as global alignment when calculating score for the similarity searches. Local alignment is done when comparing a single sequence to an entire database. While global alignment is used to align entire sequence that is as long as the sequence length is. So in contrast to local and global alignment BLAST takes over more advantage in comparison to FASTA as BLAST does local and global alignment according to user need.[4]

One of the most important point to accomplish the more importance of BLAST over FASTA is that recently in 2009,a new approach is been introduced called Pattern Hunter which is applied on the current features of BLAST and make it more versatile to particularly human genome for DNA purpose. In Pattern Hunter, a filtering technique is used for the enhancement of sequence comparison speed. The Pattern Hunter discovers short word matches under spaced form. A spaced form is represented as a binary character in 0 and 1.bit 1 at any position means a match is detected and bit 0 at any position means a mismatch is detected. This pattern Hunter eliminate the UN required DNA sequences from being executed on local alignment as well as global alignment. After this achievement BLAST becomes rapidly powerful tool for a specifically human genome world.[3]

5. CONCLUSION

BLAST are used to compare biological sequences against a public database and similarly also used for pattern matching of any unknown gene or sequences. BLAST algorithm and programs has been typically designed for speed. Hence BLAST algorithm will be extremely useful in knowing whether user query sequence is related to any other genome or proteins. [6]

FASTA algorithm is the first fast sequence searching algorithm for comparing a user query against an available database.

Lastly I conclude that by studying the both design style and structure of BLAST and FASTA, keeping in view there pros and cons and also by analyzing there statically results by passing queries and analyzing there outputs, I conclude that BLAST algorithm and BLAST tools is much far better and user friendly for an ordinary person who has no relationship to a biological environment. and finally by preparing a comparative study of both BLAST and FASTA, I personally got to know that BLAST tools and techniques is more adaptable for an ordinary user and secondly enhancement to BLAST features and versions it also add flexibility in BLAST to be use variety of human biological relations and across worldwide.

REFERENCES

[1] Jian Ye, Scott McGinnis and Thomas L. Madden “BLAST: improvements for better sequence analysis” National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health: March 20, 2006.

[2] Ke Chen, Kun She, William Zhu, Qing-xin Zhu “Optimal and Adaptive Pair wise DNA Sequence Correlation Analysis in Natural Gradient” School of Computer, Univ. of Electron. Sci & Tech of China, Chengdu 610054; 2008,IEEE.

[3] A. R. M. Nordin, M. S. M. Yazid and A. Aziz” A Guided Dynamic Programming Approach for Searching a Set of Similar DNA squences” Faculty of Informatics, Universiti Darul Iman Malaysia, 21300 K Terengganu MALAYSIA;2009,IEEE.

[4] Piet Jan Andree, Mark F. Harper, Stéphane Nauche, Robert A. Poolman, Jo Shaw, Joop C. Swinkels , Sally Wycherley” A comparative study of patent sequence databases” Scientific Information and Library Services, Sanofi-Aventis Recherche et Developpement, 13 Quai Jules Guesde, Vitry sur Seine 94403, France:2008.

[5] Enis Afgan · Purushotham Bangalore” Exploiting performance characterization of BLAST in the grid” Springer Science+Business Media, LLC, 2010.

[6] Dong-bu-bo,Lin Xu,Ning-Hui Sun,Gaung Ming Tun” Improvement of performance of MegaBLAST for DNA

sequence Algorithm" Institute of computing technology,Feburary,2006.

[7] Maria Mirto, Sandro Fiore, Italo Epicoco, Massimo Cafaro, Silvia Mocavero, Euro Blasi, Giovanni Aloisio" A Bioinfomatics Grid Alignment Toolkit" University of Salento, Lecce & SPACI Consortium, Italy, February 2008.

[8] Andre Silvanovich, Gary Bannon, Scott McClain" The use of E-scores to determine the quality of protein alignments" Monsanto Company, Regulatory Product Characterization Center, 800 North Lindbergh Blvd., St. Louis, MO 63167, USA, February 2009.

[9] Lorenza Bordoli, Swiss Institute of Bioinformatics" Similarity Searches on Sequence Databases: BLAST, FASTA" EMBnet Course, Basel; October 2006.

[10] Shannon Steinfadt, Dr. Johnnie W. Baker" SWAMP: Smith-Waterman using Associative Massive Parallelism" Department of Computer Science, Kent State University, Kent, Ohio 44242 USA; 2008, IEEE.