APPLICATION NOTES

Genomics & Informatics- Vol. 4(4) 173-181, December 2006

A Pattern Summary System Using BLAST for Sequence Analysis

Han Suk Choi[1]*, Dong-Wook Kim[2] and Tae W. Ryu[3]

[1,2]Deaprtment of Multimedia Engineering, Mokpo National University, 534-729 Dorim-ri, Chungkye-myun, Muan-Gun, Chonnam, Korea, [3]Department of Computer Science, California State University-Fullerton, Fullerton, CA, USA

Abstract

Pattern finding is one of the important tasks in a protein or DNA sequence analysis. Alignment is the widely used technique for finding patterns in sequence analysis. BLAST (Basic Local Alignment Search Tool) is one of the most popularly used tools in bio-informatics to explore available DNA or protein sequence databases. BLAST may generate a huge output for a large sequence data that contains various sequence patterns. However, BLAST does not provide a tool to summarize and analyze the patterns or matched alignments in the BLAST output file. BLAST lacks of general and robust parsing tools to extract the essential information out from its output. This paper presents a pattern summary system which is a powerful and comprehensive tool for discovering pattern structures in huge amount of sequence data in the BLAST. The pattern summary system can identify clusters of patterns, extract the cluster pattern sequences from the subject database of BLAST, and display the clusters graphically to show the distribution of clusters in the subject database.

Keywords: Pattern Finding, BLAST, Sequence Analysis, Protein, DNA

Introduction

Pattern finding is one of the important tasks during DNA sequence analysis. The information embedded in the proteins and DNA sequences is described by sub-sequences or elements in the strands. Researchers found that some sub-sequences have higher frequency than others in a protein or DNA sequence. These sub-sequences usually correspond to functionally or structurally important elements in proteins or DNA sequences, and are preserved better than other sequences during the evolution time.

The technique and algorithms of pattern findings that can quickly find out those significant elements or the meaningful patterns from huge amount of sequence data will be very useful for the biological research in many areas such as multiple sequence alignment, protein structure and function prediction, characterization of protein families, promoter signal detection, and so on. Alignment is the widely used technique for finding patterns in sequence analysis (Feng et al., 1996; Hughey *et al.*, 1996).

BLAST (Basic Local Alignment Search Tool) is one of the most popularly used tools in bio-informatics to explore available DNA or protein sequence databases. BLAST can be obtained from the NIH's National Center for Biotechnology Information (NCBI) at no charge. It consists of five programs: blastp, blastn, blastx, tblastn, and tblastx (University of Oxford web site,1997). BLAST is very popular not only because it's wide availability, but also it allows researchers to use the databases containing previously characterized genes to compare their new sequences (Ostell *et al.*, 1996; Zhang *et al.*, 1997).

However, BLAST also has some disadvantages as follows : (a) BLAST lack of general and robust parsing tools to extract the essential information out from its output. (b) The standalone version of BLAST does not have the user-friendly graphics interface. For running standalone BLAST, users have to enter the command and all option argument on the command line. (c) BLAST may generate a huge output for a large sequence data that contains various sequence patterns (e.g , small character patterns, large chunk of patterns, redundant patterns, etc.). BLAST does not provide a tool to summarize and analyze the patterns or matched alignments in the BLAST output file.

This paper presents a pattern summary system which is a powerful and comprehensive tool for discovering pattern structures in huge amount of sequence data. In addition, the pattern summary system will meet the following key objectives: (a)Create a windows environment for BLAST programs such as "formatdb" and "blastall" so that user can easily choose the optional arguments and run them. (b)Create a flexible and scalable BLAST output parser to extract pattern information. (c)Create a module that can Identify clusters of patterns, extract the cluster pattern sequences from the subject database. and display

*Corresponding author: E-mail chs@mokpo.ac.kr,
Tel +82-61-450-2354, Fax +82-61-450-2358
Accepted 15 Dec 2006

```
BLASTN 2.2.5 [Nov-16-2002]

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.

Query= Test1
         (560 letters)

Database: E:\blast\ecoli.nt
           400 sequences; 4,662,239 total letters
```

Fig. 1. BLAST information and database

```
>gi|1786181|gb|AE000111.1|AE000111 Escherichia coli K-12 MG1655
           section 1 of 400 of the complete genome
          Length = 10596

 Score =  589 bits (297), Expect = e-168
 Identities = 315/324 (97%)
 Strand = Plus / Plus

Query: 237 aggtaacggtgcgggctgacgcgtacaggaaacacagaaaaaagcccgcacctgacagtg 296
           ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 237 aggtaacggtgcgggctgacgcgtacaggaaacacagaaaaaagcccgcacctgacagtg 296
....
....
Query: 537 cgaacgtatttttgccgaactttt 560
           |||||||||||||||||||||||
Sbjct: 537 cgaacgtatttttgccgaactttt 560
```

Fig. 2. List of Hits

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

gi|1786181|gb|AE000111.1|AE000111 Escherichia coli K-12 MG1655 s...   589   e-168
gi|2367252|gb|AE000440.1|AE000440 Escherichia coli K-12 MG1655 s...    32   0.51
```

Fig. 3. Significant alignments

the clusters graphically to show the distribution of clusters in the subject database. (d)Build a framework to integrate existing tools for finding pattern structures. This framework will have a user-friendly interface and easy to expand it.

BLAST for Pattern Findings

There are two ways to use BLAST. one way is go to some public website such as http://www.ncbi.nlm.gov/BLAST/ to use their services. In this way, users can their own query sequences to BLAST against the public gene database. The general steps for doing this can be summarized as follows: user first open the website, then select the appropriate program from blastp, blastn, blastx, tblastn, and tblastx. After gave their own query sequences in a text field by copy and paste, the user can setup some necessary options for advanced blasting and the format of BLAST output file. By clicking the submit button, the query sequences will be uploaded to the web site and the searching result will be returned back to the user and displayed on the browser.

The second way is to download the stand alone BLAST and install it on the local machine. In this way, users can create their own database and run the BLAST anytime. The input files for BLAST include subject sequence database and the query sequences file both in text file format.

The output of BLAST can be divided into three parts: information about the program, a list of the hits, and alignment results. The following is an example result generated by blastn which compares a nucleotide query sequence against a nucleotide sequence database.

Part I: Figure 1 gives the information about the program and the database that was searched. In this example, program is BLASTN 2.2.5 and the database Is

"ecoli.nt". The database has 400 sequences and total length of the database is 4,662,239. "Query= Test1" represents that the query ID is Test1.

Part Ⅱ: Figure 2 shows a list of the 'hits'. A hit can be considered as a pattern found. In this example, the database name and the first part of the sequence description are given for each hit as well as the "Score" and the "Expect Value".

Part Ⅲ. Figure 3 shows the alignments between query sequence and the database sequences. In this example, the "Score" is the sum of the scoring matrix values in the segment pair being displayed. From the sequences displayed, the start point and end point of this hit both in query sequence and database sequence are given as well as the sequence of the hit.

The format of the summary file can be described as follows: all data of one hit are written in one line and separated by ",", and no white space in between allowed. The data from left to right are database name, database length, hit start position in the database, hit end position in the database, query ID, query length, hit start position in the query, hit end position in the query, strand direction, evaluation of the alignment, actual hit number, length of hit and the length of gap. The following is an example:

CHROM_III,15279300f94216,895826,TC1,1610,1,1610.-1,0.0,1609,1611,1
CHROM_III,15279300,12520644,12522254,TC1,1610,1,1010,1,0.0,1607,1611,1
CHROM_III,15279300,784389,786000,TC1,1610,1,0.0,1608,1612,2
CHROM_III,15279300,1935481,1937092,TC1,1610,1,1610,1,0.0,1606,1612,2
CHROM_III,15279300,6571164,6572774,TC1,1610,1,1610,-1.0.0,1604,1611,1

Pattern Summary using Clustering Approach

The purpose of BLAST is to find the matched sequence segments (or hits) as many as possible by comparing the query sequences with the subject sequence database. Each individual hit does not give the useful information to the researcher, but if we focus on the distribution of the hits, we may find some significant messages of life. One way to handle this is to use "Clustering Approach". This process can be started after the BLAST output file is parsed and all the necessary information of the hits for clustering are saved in the summary file (Ben-Dor *et al.*,1999).

There are some steps for summarizing patterns from BLAST output file:

Step 1 : Identify all clusters of hits

Typically the BLAST hits come in clusters. Finding and analyzing all those clusters of hits can help researchers to discover the secret of life. A cluster of hits is a group of close by or overlapping hits that belong to the same subject and query sequences. For finding the boundary of a cluster, there are three possible cases that can help to determine if a hit is belong to a cluster or not. To better explain each case, let Hi.s and Hi.e be the starting position and the ending position of a hit Hi and Hj.s and Hj.e be the starting position and the ending position of a hit Hj respectively.

Case 1: Overlapped Hits

The hits can be overlapped in the following fashions:

Hi.s----Hj.s--Hi.e--Hj.e

Two hits, Hi and Hj are overlapped since the starting position of hit Hj is inside of the hit Hi. In this case, the boundary of the cluster is defined as Hi.s ~ Hj.e. The maximum length of the overlapping can be decided by users. Another kind of overlapping is shown below.

Hi.s----Hi.s---Hj.e-----Hi.e

A hit, Hj is located in another hit Hi. In this case, the boundary of the cluster is defined as Hi.s~Hi.e.

In both cases in the above, to find a cluster, we simply look for the min. position for the left boundary and the max. position for the right boundary of the cluster from all the records belong to the cluster. The main reason behind this decision is based on the nature of *Transposable Elements(TEs) such as frequent repeated* sequences.

Case 2: Isolated Hits

The case of isolated hits can be best shown in the example below:

Hi.s-------Hi.e---Hj.s------Hj.e

The two hits are considered isolated if length(Hi.e, Hj.s)>= gap, where length(Hi.e, Hj.s) is the length of sequence between the neighboring hits, Hi and Hj; gap is a integer assigned by the user, or some thing related to the length of the query sequence. In this case, we

identify two separate clusters, Hi.s ~ Hi.e for the hit Hi and Hj.s ~ Hj.e for the hit Hj.

Case 3: Isolated but closely Neighbored Hits

Hi.s-------Hi.e--Hj.s-------Hj.e

The neighboring hits are considered too close and will be consolidated if length(Hi.e,Hj.s) < gap. In this case, the boundary of a cluster is defined as Hi.5 ~ Hj.e. However, if the length of the cluster is longer than 1.2 times of the length of the query sequence, we also identify two separate clusters, Hi.s ~ Hi.e for the hit Hi and Hj.s ~ Hj.e for the hit Hj.

Step 2: Determine the major strand

For each hit cluster, determine the direction of the hit cluster based on the value of the attribute, strand (1 represents a hit in the same direction, e.g., +/+ hit, -1 represents a hit in the reverse direction, e.g., +/- hit). Since there are multiple hits in each cluster, we use the direction from the record with the longest hit (e.g.. biggest hit_len value) in this paper.

Step 3: Extract the sequences for each cluster

The next step is to extract the corresponding sequences from database files, for each cluster identified in step 1 and put them in a FASTA format. A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column. For the cluster sequence with strand value, -1 (e.g., +/- hit), compute the reverse complementary sequence. For example, if the extracted sequence for a cluster with strand value. -1 is AAATTTCG, then the reverse complementary of this sequence is: CGAAATTT

Step 4: Store the extracted sequence for each cluster in a FASTA format.

Interface Design of the Pattern Summary System

The goal of the pattern summary system is to provide a powerful and comprehensive tool for pattern structure discovery using BLAST and other alignment tools such as ClustalX. By using this system, researchers can identify interesting patterns in the BLAST output file for the given sequences, extract the significant cluster sequences from the subject sequence database, display the clusters of hits in a nicely formatted graph, and align the clusters of hits.

Interface Design and Major Components

This paper represents a framework with user-friendly graphical interface for the users. Inside the framework, five modules, FORMATDB, BLASTALL. PARSER, CLUSTER and ALIGNMENT will be integrated. FORMATDB and BLASTALL are two functions of BLAST that can be obtained from NCBI website. ALIGNMENT calls the multiple sequence alignment program CLUSTX, which is the Windows version of ClustalW to align the matched sequences. CLUSTX provides an integrated environment for performing multiple sequence and profile alignments and analyzing the results. PARSER and CLUSTER are the main tasks of this study and will be newly developed. PARSER allows user to parse the output file of BLAST and save the result in different format. CLUSTER analyzes the output file of BLAST based on the result of PARSER.

The steps typically involved in sequence analysis using BLAST are shown as follows:

Step 1: Format the subject database

For formatting subject database the program "formatdb" can be used "formatdb" must be run to reform the subject database in certain format before running "blastall" "formatdb" is a DOS mode or Shell mode program that can be run by typing formatdb on command line followed by the optional arguments.

Step 2: Blast the query against subject database

For blasting the query against subject database, the program "blastall" of BLAST can be used. Like "formatdb", "blastall" is also a DOS mode or Shell mode program and also should be invoked in a window that has all possible options for running it.

Step 3: Parse the output of BLAST

For parsing the output file of BLAST, a parsing module is needed. This module takes BLAST output file as input, and parse out the hits information and the sequences of the hits. This module should give the choice to the user to select the columns that should appear in the result, and the sequences of hits are saved in FASTA format.

Step 4: Clustering sequence patterns

For finding, extracting and displaying clusters of patterns, a clustering module is needed. This module takes the

output file of the parsing module with the delimiter "," as input. It also asks the user to enter the gap length between two clusters in order to determine cluster boundary and answer the question that if he or she wants to extract cluster sequences from database after the clusters are found.

Step 5: Sequence Pattern Alignment

For aligning cluster sequences, the open source program "ClustalX" can be used. ClustalX is a windows application and can be downloaded from many web sites. It provides a nice interface for user and does not require arguments when start It.

Major Components

The Pattern Summary System consists of six main components; Main Control Module, FormatDB, BLAST, BLAST Parser, Cluster Patterns, and Alignment Module. The Main Control Module shown Figure 4 is the driver of this whole system. It works as a frame that integrates other five modules: "FORMAT BLAST DATABASE", "BLAST", "BLAST PARSER", "CLUSTER", "ALIGNMENT".

FormatDB takes sequence database as input as well as other options for calling the external program "formatdb". These extracted files have the same name as the name of subject sequence database, but the extensions are different. Figure 5 shows the FORMAT SUBJECT DATABASE FOR BLAST GUI interface.

BLAST takes the formatted target sequence database generated by formatdb as input as well as other options for calling the external program "blastall". By click on the button "Format Database", the blast output file will be created. Figure 6 shows the full options of "blastall" GUI interface.

BLAST Parser takes the BLAST output file as input. By click on the button "Parse", there are two output file will be created, summary file that consists of information of each hits, and the sequences file. Figure 7 shows the

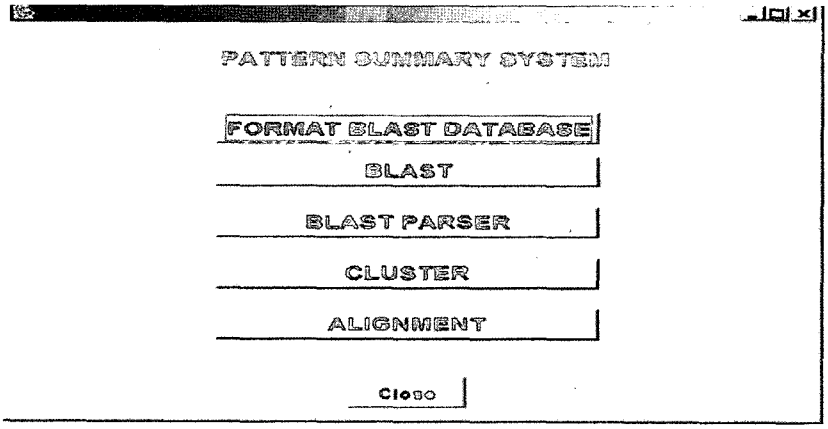

Fig. 4. Main Control Module of the Pattern Summary System

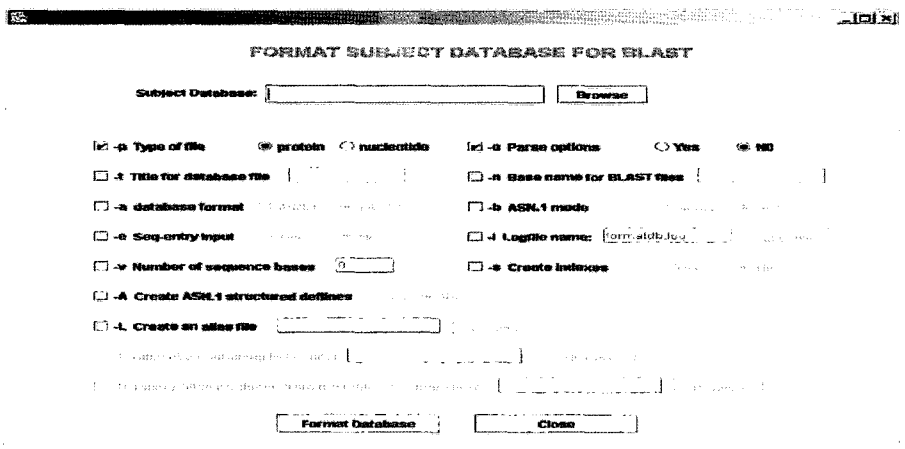

Fig. 5. Format Subject Database Interface for BLAST

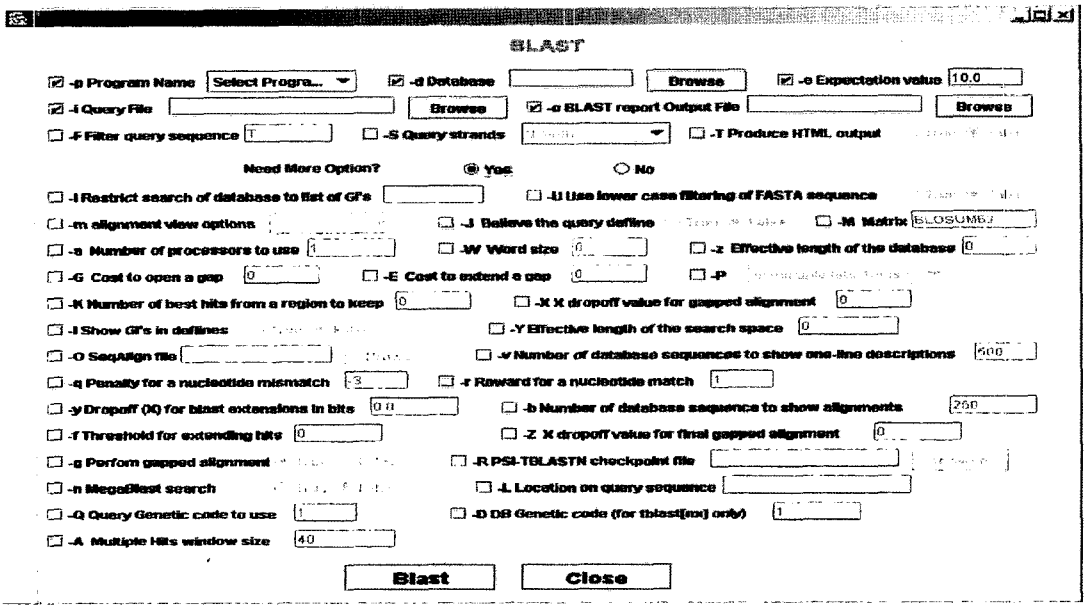

Fig. 6. BLAST Interface of the Pattern Summary System

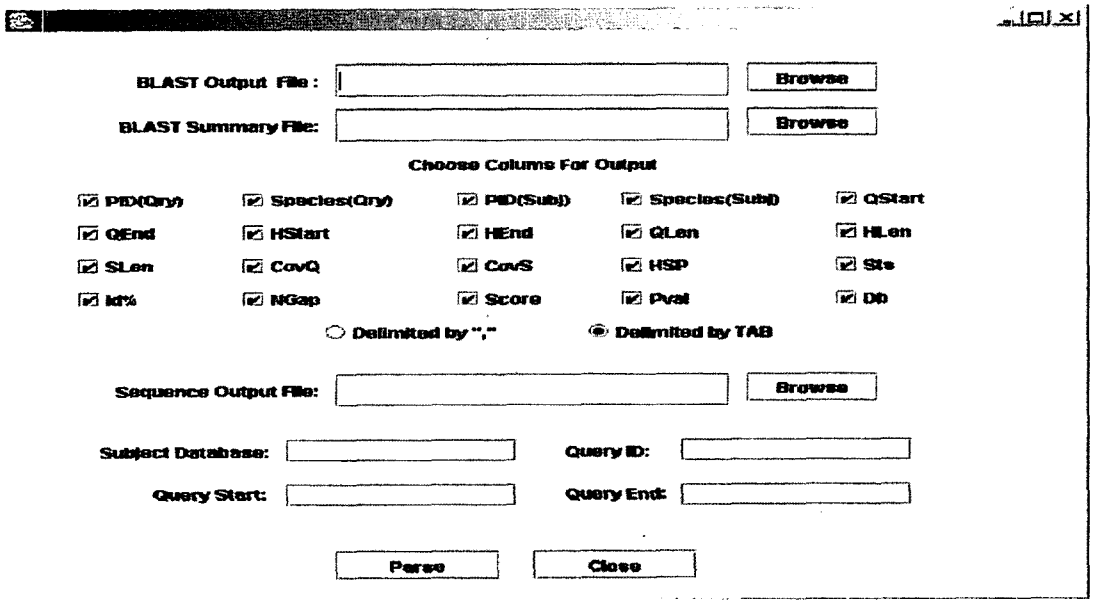

Fig. 7. BLAST PARSER Interface of the Pattern Summary System

BLAST Paser interface.

Clustering Patterns Module takes the summary file of hits generated by BLAST parser as input as well as the length of the gap between clusters determined by the user, and the coefficient for the longest length of cluster. The gap length can be the number of characters between two clusters, or a percentage of record length. The coefficient is used to limit the length of cluster. Figure 8 shows the input interface of the CLUSTER BLAST HITS. Cluster result window can call other two functions. One is "extract.pl", which is an external function written in Perl. "extract.pl" extracts the cluster sequences from subject database and save it into a text file. The other one is the internal function "Graph", which visually displays the clusters. Figure 9 is the output interface and a result of the CLUSTER LIST.

Alignment Module provides an integrated environment for performing multiple sequence and profile alignments

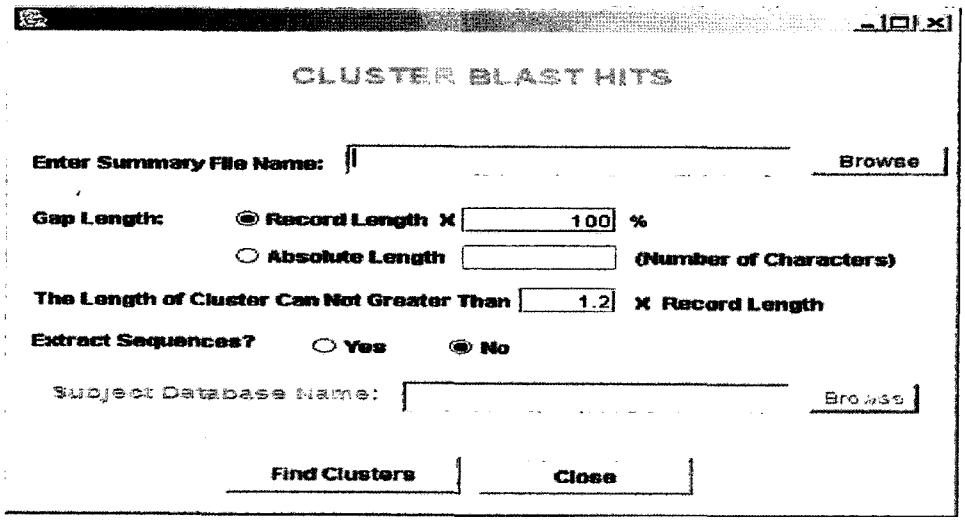

Fig. 8. CLUSTER BLAST HITS Input Interface of the Pattern Summary System

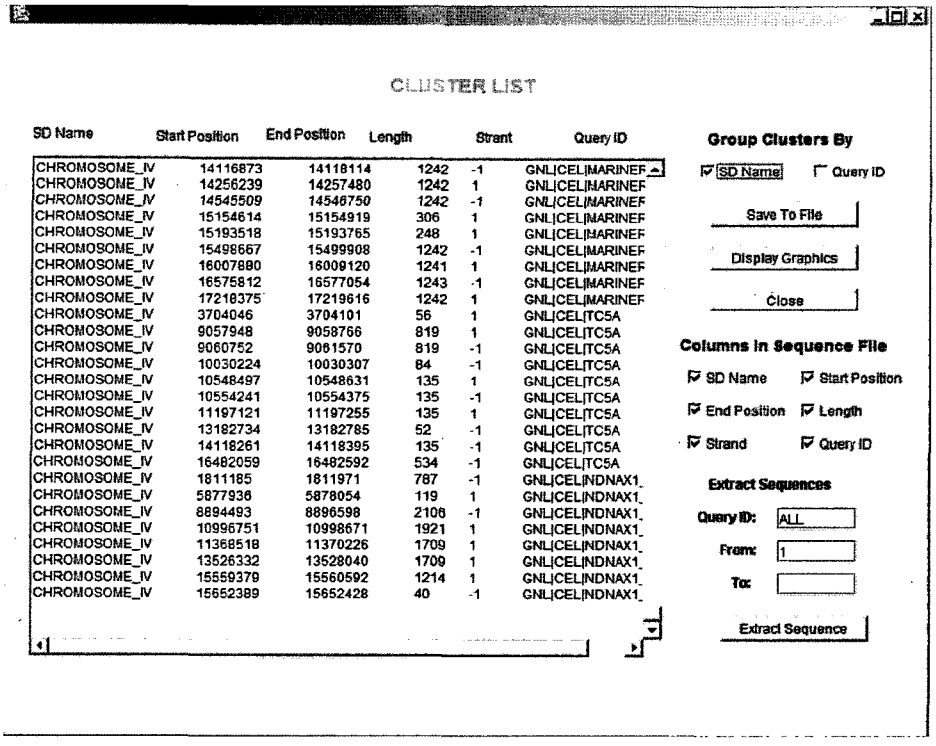

Fig. 9. CLUSTER LIST of the Pattern Summary System

and analyzing the results. The sequence alignment is displayed in a window on the screen. Figure 10 is the result of CLUSTER LIST Alignment.

A versatile coloring scheme has been incorporated allowing user to highlight conserved features in the alignment. The pull-down menus at the top of the window allow user to select all the options required for traditional multiple sequence and profile alignment. User can cut-and-paste sequences to change the order of the alignment, select a subset of sequences to be aligned, and can select a sub-range of the alignment to be realigned and inserted back into the original alignment. Alignment quality analysis can be performed and low-scoring segments or exceptional residues can be highlighted.

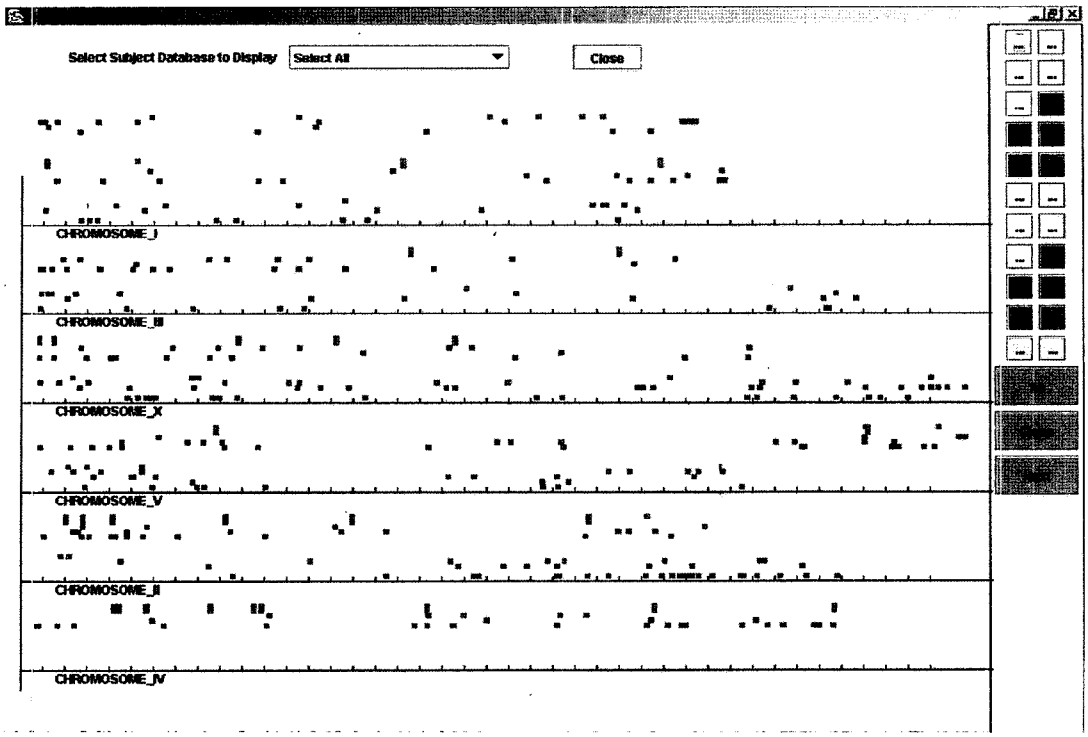

Fig. 10. Alignment of Cluster Sequence of the Pattern Summary System

Implementation of Pattern Summary System

The pattern summary system implemented in this study is a platform independent. It can be installed and run on most commonly used operating systems such as Windows, Linux and Unix because it was implemented by using JAVA programming language. It also have to be able to provide user friendly GUI. Moreover, it has powerful API that makes the implementation much easier.

Case Study

All living organisms contain Transposable Elements (TEs), which are the repeated sequences inserted into the host genomes and can move from one position to another along the chromosomes. In order to propagate and survive, TEs have to maintain a certain level of activity. Consequently, the study on how these elements move and integrate into the genome, home the host, maintain its transposition under control and how TEs get around the regulation to insert into new locations is very important. For doing this research, the TEs have to be searched and compared with many other genome sequences. Since the length of genome sequences are extremely long, finding a pattern from genome sequences needs a lot of manpower and is very time consuming, and the searching algorithm is becoming an issue. This paper test the pattern summary system to find the clusters of transposable elements in the subject database.

Subject database

The subject database can be any sequence file in FASTA format. The subject databases we are using in this project for the purpose of testing are downloaded from the web site http://www.wormbase.org/ chromosomes/ There are total six files, CHROMOSOME_I, CHROMOSOME_II, CHROMOSOME_III, CHROMOSOME_IV, CHROMOSOME_V, and CHROMOSOME_X. We join these six files together into a file "chro.txt" so that we can do the BLAST against one subject database instead of six databases. Its total size is 102,69,453bytes.

Query sequences

The query sequences file is named "rep_autote_cel.fas". It contains 22 transposable elements.

Parse Hits

After we got all matched sequences and saved them into output file, we need to get the necessary information for finding clusters through the window "Parser". There are

total 2732 hits in the file "chros_sum".

Find Clusters

Among the 2732 hits, we found the clusters by using the module Parser. Some hits come in cluster. To identify these clusters, the module "FIND CLUSTER" can be used. As the result, there are total 845 clusters within 2732 hits.

Extract Cluster Sequences

The cluster sequence can be extracted from database through the window "CLUSTER LIST" We can extract first 30 character for all cluster sequence and save into the file. Figure 9 shows an example of the cluster list.

Display Clusters

By clicking on button "Display Graph" in the window "Cluster List", the GUI of class Graph will be opened as Figure 10. In the Figure 10, all the cluster we found in the BLAST output file will be displayed on the screen with different color depend on which query sequence it belongs to. The information of each cluster can be found by making a mouse click on its color square image.

Conclusion

Sequence analysis is one of the important areas in bio-informatics research. Many different tasks involved and approaches have been proposed for sequence analysis. One of the most important tasks involves the pattern discovery. BLAST is the most popularly used tool for pattern discovery in sequences. However, BLAST may generate huge amount of results containing hits (or patterns), particularly when the input sequence file is big. Interpreting the huge size of BLAST output file is not trivial.

The pattern summary system provides a user-friendly environment that integrates BLAST, tools for parsing the output file generated by BLAST, finding clusters of hits from the parsed output file, and alignment tool. Through this system, users can utilize features supported by several tools from the beginning to the end of pastern discovery without switching from one program to another. Another significant feature of this paper is that it can display clusters of hits with different colors on the screen. User can display different clusters of hits together, or display one cluster at a time. It also can display clusters of hits for an individual subject database or for all databases. With these functionalities, user can visually observe the pattern distribution in a database and easily to catch significant phenomenon.

References

Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999). Clustering gene expression patterns. Journal of Comptu. *Biol.* 6, 281-297.

Feng, D. F. and Doolittle, R. F. (1996). Progressive alignment of amino acid sequences ad construction of phylogenetic trees from them. *Methods Enzymol.* 266, 368-382.

Hughey, R., Krogh, A., Barrett, C., and Grate, L. (1996). SAM: Sequence alignment and modeling software. University of California. Baskin center for Computer Engineering and Information Sciences (http://www.cse.ucsc.edu/research/compbi/papers/sam_doc/sam_doc.html).

Sequence Analysis. (2002). The Kimmel Cancer Center NCI-designated. (http://www.kcc.tju.edu/ Science/whatIs/what_is_sequence_analysis.htm).

Holguin, G., and Patten, C. (2000). Finding Patterns in Biological Sequences.

Ostell, J. M. (1996). The NCBI software tools. In Nucleic Acid and Protein Analysis: A Practical Approach, M. Bishop and C. Rawlings, Eds. (IRL press, Osford), p.31-43.

Subramaniam, S., and Pevzner, P. (2002). Heuristic Alignment Program for Database Search. (http://genome.ucsd.edu/classes/be202/html/ part5.html).

Zhang, J. and Madden, T. L. (1997). Power BLAST: A new network BLAST application for interactive or automated sequences analysis and annotatuon. *Genome Res.* 7, 649-656.