American Journal of Bioinformatics 2 (2): 15-23, 2013
ISSN: 1948-9862

doi:10.3844/ajbsp.2013.15.23 Published Online 2 (2) 2013 (http://www.thescipub.com/ajb.toc)

AN EFFICIENT DISTRIBUTED BIOINFORMATICS COMPUTING SYSTEM FOR DNA SEQUENCE ANALYSIS ON ENCODING SYSTEM

Mohammad Ibrahim Khan and Chotan Sheel

Department of Computer Science and Engineering (CSE),
Chittagong University of Engineering and Technology (CUET), Chittagong-4349, Bangladesh

Received 2013-11-06; Revised 2013-12-20; Accepted 2013-12-20

ABSTRACT

This study provides an effective design of search technique of a distributed bioinformatics computing system for analysis of DNA sequences using OPTSDNA algorithm. This system could be used for disease detection, criminal forensic analysis, gene prediction, genetic system and protein analysis. Different types of distributed algorithms for the search and identification for DNA segments and repeat pattern in a given DNA sequence are developed. The search algorithm was developed to compute the number of DNA sequence which contains the same consecutive types of DNA segments. A distributed subsequence identifications algorithm was designed and implemented to detect the segment containing DNA sequences. Sequential and distributed implementation of these algorithms was executed with different length of search segments patterns and genetic sequences. OPTSDNA algorithm is used for storing various sizes of DNA sequence into database. DNA sequences of different lengths were tested by using this algorithm. These input DNA sequences varied in size from very small to very large. The performance of search technique distributed system is compared with sequential approach.

Keywords: Distributed Bioinformatics System, DNA Sequence, Search Segments, Identify DNA Sequences, Reported Gene Sequences

1. INTRODUCATION

Distributed Computing (DC) provides a cost effective frame work with efficient execution of a solution on multiple computers connected by a network. For Distributed Computing (DC), large tasks are divided into smaller problems which can then be executed on multiple computers at the same time independent of each other. The task must be broken up into independent problems to minimize inter-computers communication; otherwise distributed computing will not be effective. Over the past few years, the intermixing of computer science and the complexity of biology has lead to the prosperous field of bioinformatics (Sheel *et al*., 2013). Advances in molecular biology and technology for research have facilitated the process of sequencing of large portions of genomes in various species. Today computers have made medical research more efficient and accurate, by using parallel and distributed computers and complex biological modeling. Bioinformatics, is one of the newer areas and has opened our eyes to a whole new world of biology (Sheel *et al*., 2013; Kumar *et al*., 2007).

The fusion of computers and biology has helped scientists learn more about species, especially humans (Baxevanis and Ouellette, 2001; Durban *et al*., 1998; Gusfield, 1997). With the aid of the computers, we have learned a great deal about genetics, but there still stand many unanswered questions, that are being researched today. DNA sequence analysis can be a lengthy process ranging from several hours to many days. This study

Corresponding Author: Chotan Sheel, Department of Computer Science and Engineering (CSE), Chittagong University of Engineering and Technology (CUET), Chittagong-4349, Bangladesh Tel: +88-01818981326

builds a distributed system that provides the solution for many bioinformatics related applications.

The overall goal of this study is to build a Distributed Bioinformatics Computing System for genetic sequence analysis of DNA. This system is capable of searching and identifying gene patterns in a given DNA sequence. For the purpose of computing we stored a large no. of DNA sequence using OPTSDNA algorithm (Sheel *et al*., 2013) and segments is divided two to six consecutive nucleotide (Khan and Sheel, 2013). The system was tested for its correctness and efficiency. Different lengths of DNA sequences were used for the consecutive and non-consecutive pattern search to compare the system's response time obtained using single and multiple computers (OMIM, 2005). In addition, different lengths of DNA sequences were also used for the pattern identification to compare its response time observed using a single computer and multiple computers. Several different distributed implementations of search algorithms have been reported in the literature. The characteristics of some of those distributed algorithms are listed in **Table 1** (please see supplementary materials).

It can be observed that the most of the existing approaches require high performance parallel processors and are not implemented on loosely coupled distributed network. Moreover, most of them require specialized programming language for their implementation on these parallel processors.

The specific objective of the proposed distributed algorithm for analysis of DNA sequences are:

- Develop an effective distributed DNA sequence analysis algorithms for pattern matching of DNA Gene sequence and sub-sequences identification
- Implement them on loosely coupled distributed network such as regular local area network and wide area network using standard programming language

This study is organized in four sections. Section 1 discusses background with the applications of the distributed algorithms. Section 2 discusses the methodology with the design of distributed algorithms for DNA analysis and distributed algorithms. Section 3 discusses the results and discussion and conclusions included in section 4.

1.1. Application of the Proposed Distributed Algorithm

This distributed Bioinformatics system developed in this study could be used for disease detection, criminal forensics analysis, genetics systems and protein analysis. Di-let, Triplet, Tetra-let, Pentad-let, Hexed-let repeats formally known as a Di-nucleotide, Tri-nucleotide, Tetra-nucleotide, Pent nucleotide, Hex nucleotide. Repeat occurs when two, three, four, five and six consecutive nucleotides are repeated within a specific region of DNA sequence. These repeats can occur within or between genes. These consecutive repeats are frequently located in genes that encode transcription factors and which are active in the organism development process. Extensive Di-let, Triplet, Tetra-let, Pant-lets, Hex-let repeats are found when a mutation occurs in a gene. This mutation increases the number of occurrences of a particular nucleotide which can lead to a number of neurodegenerative diseases. These diseases include, Huntington's Disease (HD), Fragile X Syndrome, Kennedy's Disease, Myotonic Dystrophy, Spinocerebellar Ataxia Type 1 (SCA1), Dentatorubral Pallidoluysian Atrophy (DRPLA) and Fragile X E mental retardation (FRAXE). In Kennedy's Disease, Huntington's disease, Spinocerebellar Ataxia Type 1 and Dentatorubral Pallidoluysian atrophy, the number of triplet repeats is quite small, in contrast to Fragile X Syndrome, Myotonic Dystrophy and FRAXE, where the number of consecutive repeats may be very large, producing alleles that consist of thousands of repeats. These algorithms can help to detect Di-let, Triplet, Tetra-let, Patna-led and Hex-let repeats in gene sequence and can also search through DNA sequences to identify most frequently occurring repeats.

The proposed distributed algorithms will be able to first identify a DNA sequence Gene pattern in the DNA obtained from the crime scene and then it can search for those patterns in suspects DNA, which will be helpful for criminal investigation, Disease analysis, Gene Sequence Prediction, Human Identification. Criminal investigation can now be facilitated by the DNA forensic analysis.

Table 1. Comparative study with existing approaches

Reference	Algorithm complexity	Special purpose computer required	No. of computers required	Special language required	Useful on general network
Kumar *et al*. (2007)	O(n)	Yes	Flexible	Yes	No
Vishkin (1985)	O(n)	Yes	Not Flexible	Yes	No
Huang and Rajasekaran (2003)	O(n)	Yes	Not Flexible	Yes	No
Strumpen (1995)	$O(n^2)$	Yes	Not Flexible	Yes	No
Janiki and Joshi (2003)	O(n)	Yes	Not Flexible	Yes	No

Forensic analysis is a process by which two organism's DNA is compared with each other. DNA analysis is effective in finding criminals, because two different individuals will have different DNA sequence. In DNA analysis one can look for matching gene patterns at different locations of the suspect's DNA and the DNA obtained at the crime scene. Gene pattern matching at one, two or three locations in DNA usually aren't enough to associate a suspect with a crime, but gene pattern matches at 5 or more locations in DNA are usually good enough to identify a criminal. Experts believe that DNA forensic technology is more reliable than eyewitnesses, where the odds are fifty-fifty. In DNA analysis one can look for matches based on number of repeating patterns at different locations of the suspect's genome.

2. MATERIALS AND METHODS

2.1. Design of Distributed Algorithms for DNA Sequences Analysis

The proposed distributed algorithm is based on client server model. For distributed search and identification algorithms on DNA sequence, the proposed framework avoids duplicates computations on server machines. The two input items are provided by the user for pattern search and identification:

- The DNA sequence which is stored by OPTSDNA algorithm with extend two to six consecutive nucleotides division
- Search string DNA subsequences or identification DNA segments (Di-nucleotides to Hex-Nucleotides Segment pattern)

Using OPTSDNA algorithm, the DNA sequence is broken up in X segments where X = m * p. Here m = number of storage DNA and p = length of storage nucleotide base. Number of storage DNA is also used as number of servers used in distributed algorithm implementation and length of storage nucleotide base represents the length of pattern for search or identification. In the first step each server gets one segment of data and the required search or identification pattern for carrying out its computation as shown in **Fig. 1**. In addition, an offset value is sent to the server as well to make sure that no two servers are performing the same computation for search or identification. The individual results from each server are sent back to the clients where partial results are combined as shown in **Fig. 2**. The complete details of client and server side interaction are shown in **Fig. 3**. The actual pattern search for a DNA sequence with three servers is shown in **Fig. 4**, where each server starts the match at different Gene chromosome. Different starting point at various servers guarantees that no comparison for pattern search and identification is performed more than once on any server. The worst case complexity of this distributed search or identification algorithm is O (L/X), where L is the length of DNA sequence and X = m/p. In case of **Fig. 4** value of X = 1 because m = 3 and p = 3. That implies that complete DNA sequence is end to all three servers and the offset for starting the search or identification.

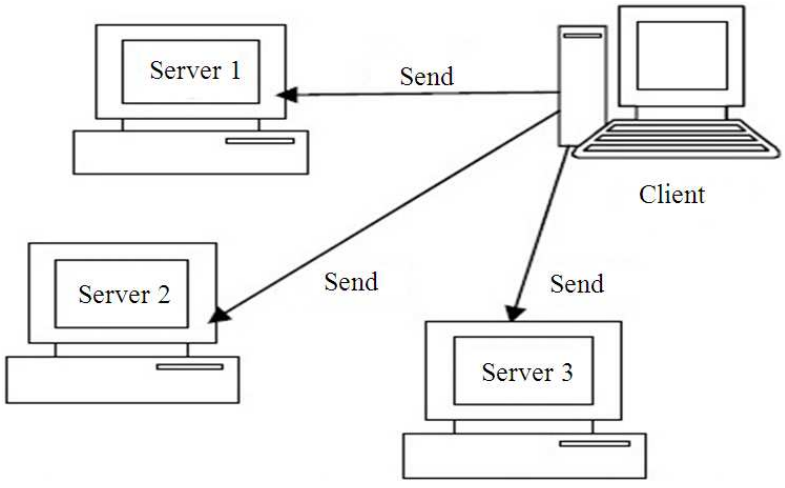

Fig. 1. Layout of the system (sending of data)

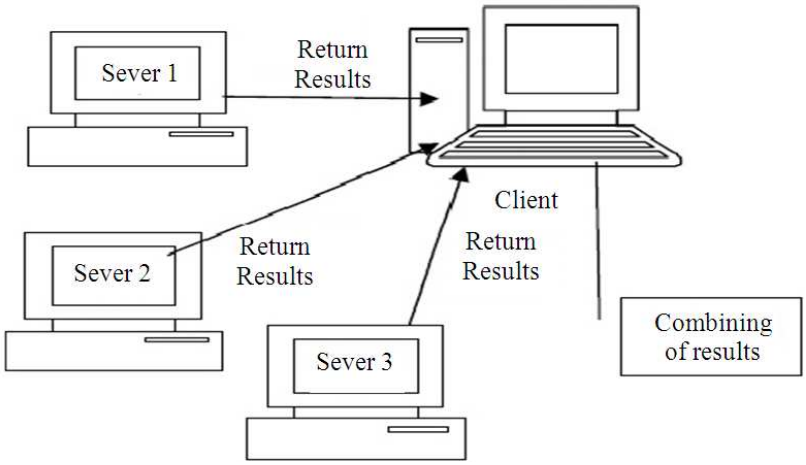

Fig. 2. Layout of system (returning results)

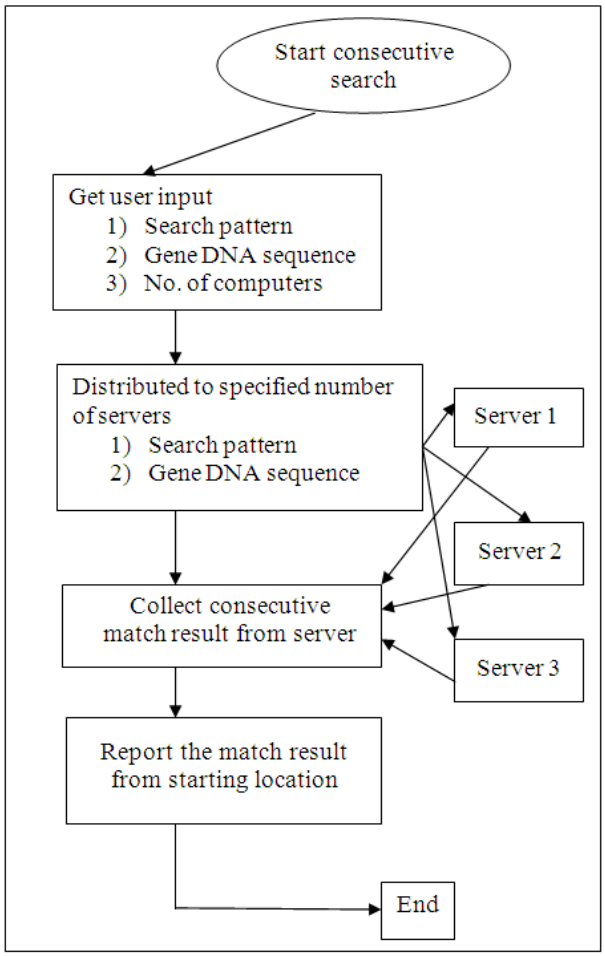

Fig. 3. Flow diagram for Clint side implementation

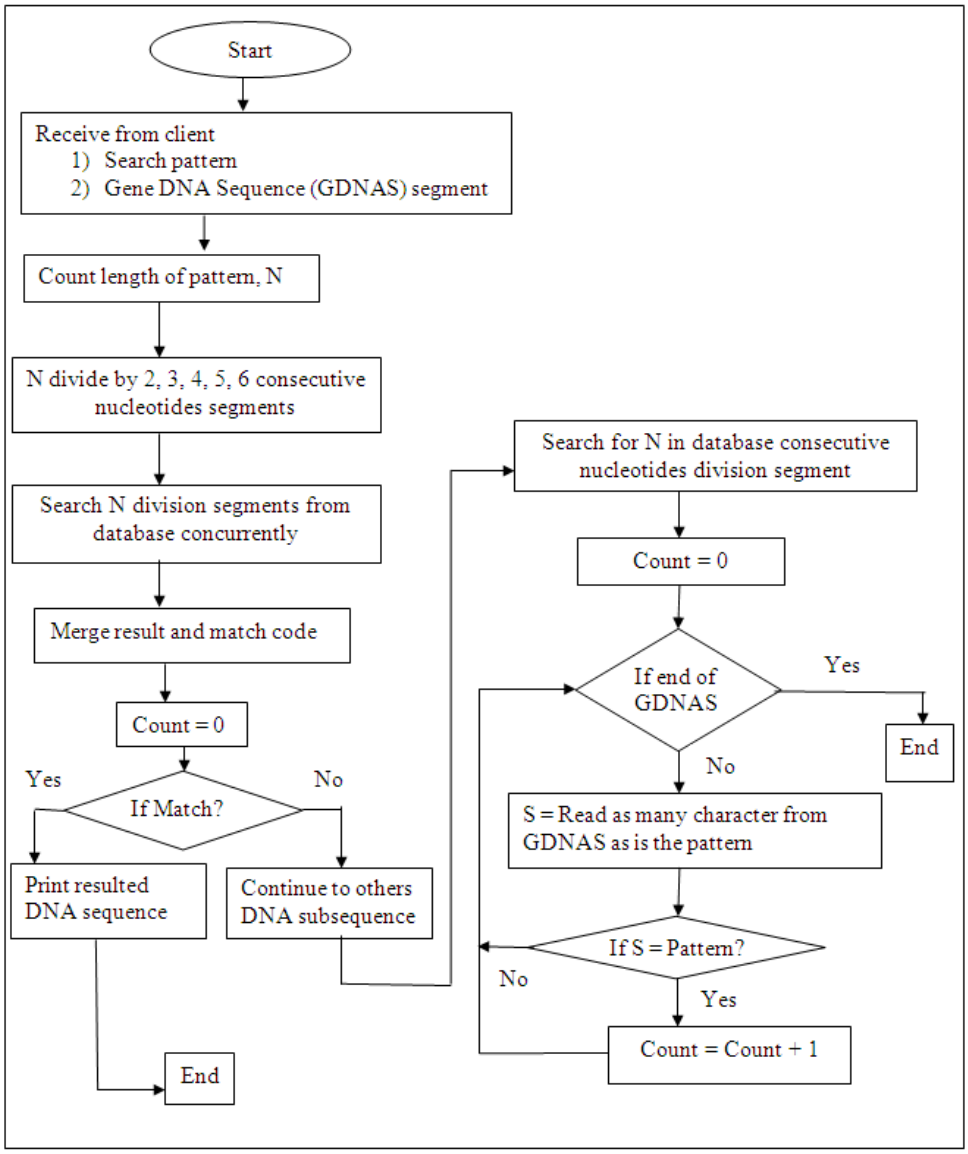

Fig. 4. Flow Diagram for consecutive search pattern from server

2.2. Implementation of Distributed Algorithms

A Dot net based client server system was developed for this project (Petroutsos, 2006; Mistry, 2008) shown in **Fig. 9 and 10** (please see in supplementary materials). The client and server side logic implementation is given in **Fig. 3 and 4**. This framework can distribute the workload across multiple servers as specified by the user. In this study, a client provides the user input from Graphical User Interface (GUI) and then send this input to one or more server computers as directed by the user (**Fig. 9 and 10**). The processing option is developed in GUI. When a client selects a processing option such as pattern identification, appropriate input for carrying out a search or identification in a DNA sequence displayed (**Fig. 9 and 10**). The client program then sends the input data to multiple servers (as specified by the user). The code at the server executes the desired algorithm and returns its results to the client. The client then

receives the results from all the servers and combines to individual results to generate a final output of the processing a shown in **Fig. 9 and 10** (please see supplementary materials).

3. RESULTS AND DISCUSSION

Sequential and Distributed versions of the algorithms were executed with different patterns of genetic sequences. These sequences were of different sizes ranging from very small to very large. The response times for sequential and distributed versions of the programs were plotted to demonstrate the effectiveness of distributed DNA sequence analysis algorithms. **Figure 5-7** shows the response time of consecutive pattern search execution on single machine and multiple machines. The execution time was calculated for DNA sequences of sizes 1 to 1000 sequences. It can be observed that execution time reduces significantly as number of servers increased. Moreover, the improvement in execution time is significant when DNA sequence size is 600 with 3 servers. **Figure 5-7** shows the response time of consecutive pattern identification execution on single machine and multiple machines. It can be observed that the execution time reduces significantly as number of servers increased.

Similar observation was made for sequential approach consecutive pattern identification algorithm execution shown in **Fig. 5-7**. **Figure 8** demonstrates how the data size affected the computation time. With a single computer the response time of each gene sequence was significantly more than that of the distributed execution using two and three servers. In addition, rate of growth of execution time is almost linear with three servers as the size of DNA sequence increases.

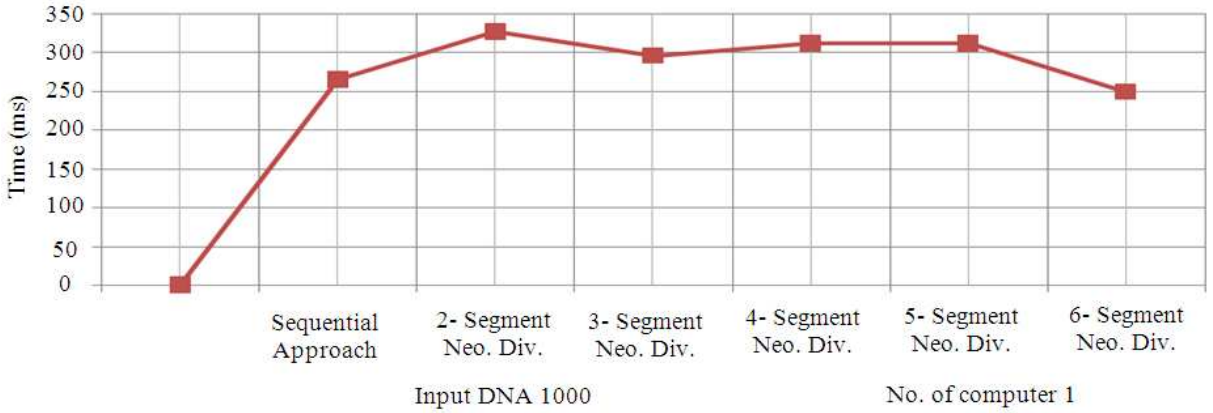

Fig. 5. Effect of data size on using single computer

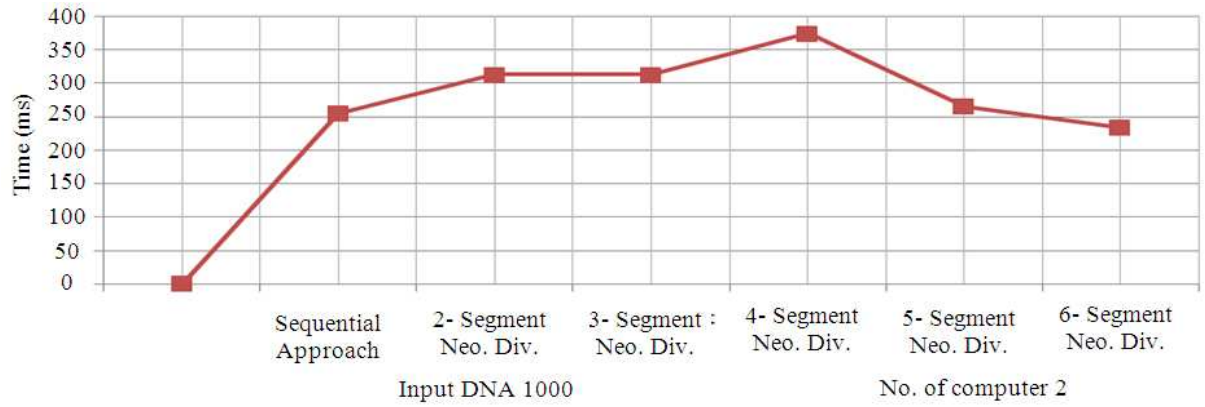

Fig. 6. Effect of data size on using two computers

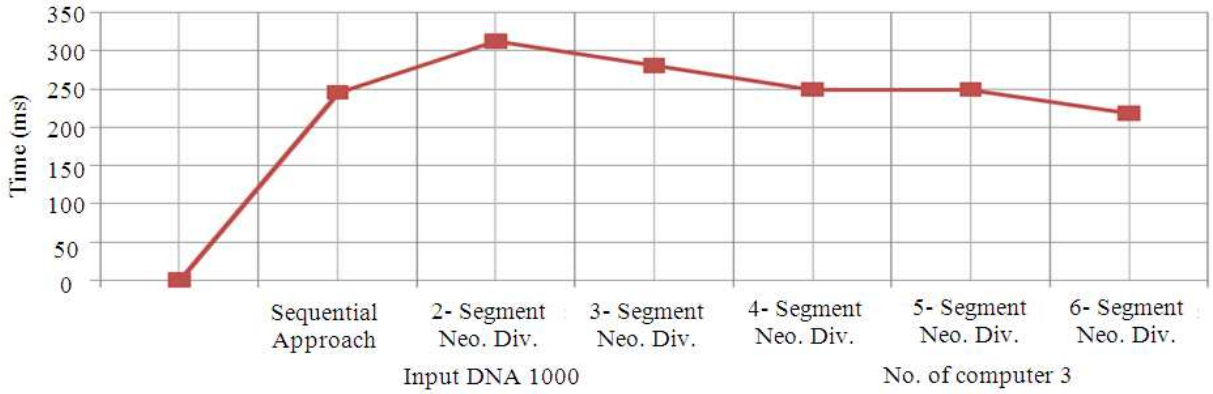

Fig. 7. Effect of data size on using three computers

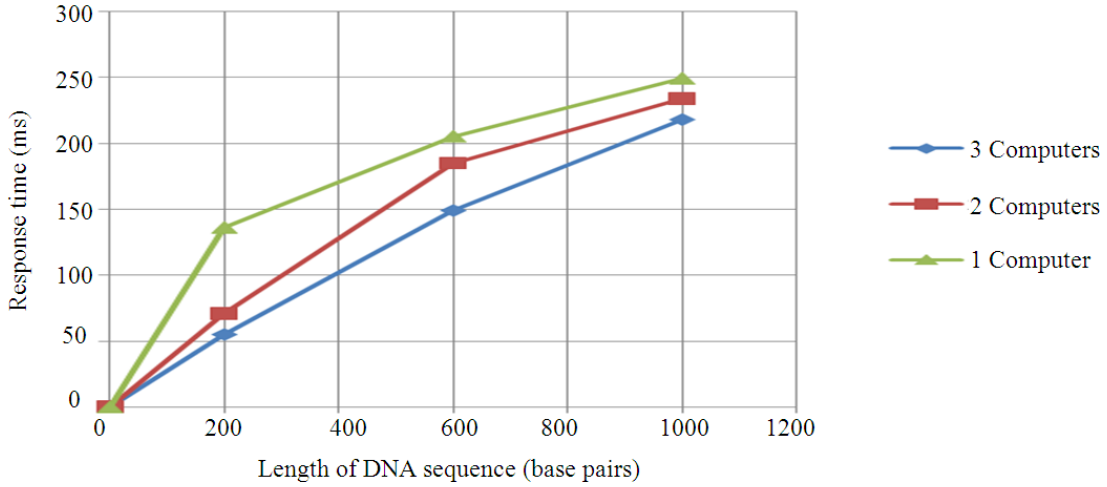

Fig. 8. Effect of data size on computation time

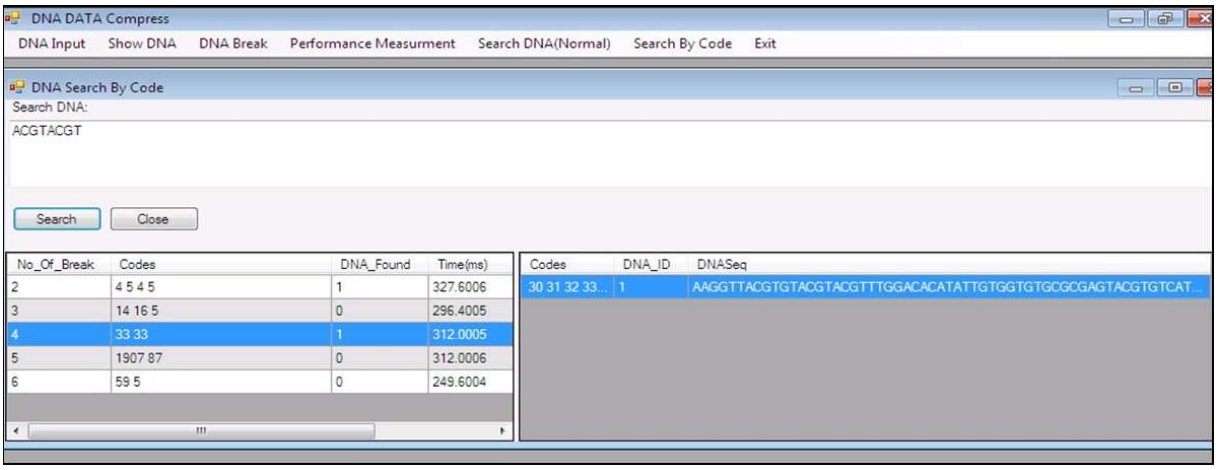

Fig. 9. Screen shot for the search process by generating code

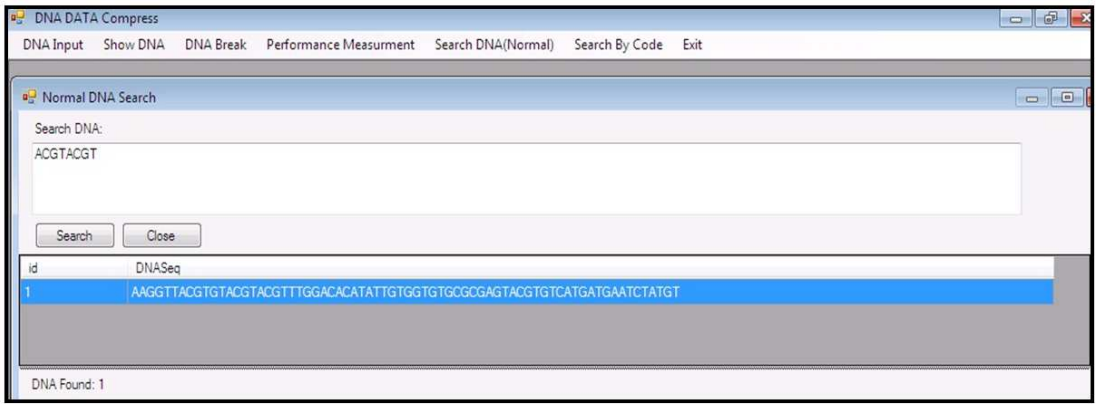

Fig. 10. Screen shot for the search process by sequential approach

4. CONCLUSION

As shown in the previous figures, it is clear that as complexity of the algorithm increases the response time also increases. The algorithm for the Pattern Identification was the most complex one and the algorithm for the pattern search was the least complex. It can be seen in **Fig. 5** the response times for the Pattern Identification were much lower compared to the other two studies shown in **Fig. 5 and 6**. This is due to the fact that more complex algorithms usually involve more steps, which increases the response time. To help get a better understanding of the effects of Distributed Systems on DNA sequences, more DNA sequences of various lengths should be tested. This would provide more data for a larger analysis. It is also recommended that the computers used in the investigation should not exceed the length of the repeat pattern that is being searched or identified, because this will not improve the response time. The complexity of our algorithm is O(n). For computing DNA sequences special purpose of computer is required. Using this algorithm no. of computer required is flexible and special language is required. Our algorithm is useful on general network. So our algorithm is more efficient then previous all. In addition, this system could be interfaced with the Internet, so that all these feature of DNA analysis are accessible to everyone via Web.

5. REFERENCES

Baxevanis, A.D. and B.F Ouellette, 2001. Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins. 2nd Edn., John Wiley and Sons, New York, ISBN-10: 0471383902, pp: 470.

Durban, R., S. Eddy, A. Krogh and G. Mitchison, 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. 1st Edn., Cambridge University Press, Cambridge, ISBN-10: 0521629713, pp: 356.

Gusfield, D., 1997. Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. 1st Edn., Cambridge University Press, Cambridge, ISBN-10: 0521585198, pp: 534.

Huang, C.H. and S. Rajasekaran, 2003. Parallel pattern identification in biological sequences on clusters. IEEE Trans. Nan Biosci., 2: 29-34. DOI: 10.1109/TNB.2003.810165

Janiki, C. and R.R. Joshi, 2003. Accelerating comparative genomics using parallel computing. Silico Biol., 3: 123-128. PMID: 12954086

Khan, M.I. and C. Sheel, 2013. OPTSDNA: Performance evaluation of an efficient distributed bioinformatics system for DNA sequence analysis. Bioinformation, 9: 842-846. PMID: 24143058

Kumar, R., A. Kumar and S. Agarwal, 2007. A distributed bioinformatics computing system for analysis of DNA sequences. Proceedings of the IEEE SoutheastCon, Mar. 22-25, IEEE Xplore Press, Richmond, VA., pp: 358-363. DOI: 10.1109/SECON.2007.342925

Mistry, R., 2008. Microsoft SQL Server 2008 Management and Administration. 1st Edn., McGraw Hill Edition.

OMIM, 2005. The OMIM gene map, NCBI.

Petroutsos, E., 2006. Mastering Visual Basic.Net. 1st Edn., John Wiley and Sons, ISBN-10: 0782152341, pp: 1184.

Sheel, C., M.I. Khan, M.I.H. Sarker and T. Alam, 2013. Algorithm for optimal storage of a distributed bioinformatics system for analysis of DNA sequences. Int. J. Computat. Bioinformat. Silico Model., 2: 106-109.

Strumpen, V., 1995. Coupling hundreds of workstations for parallel molecular sequence analysis. Software Pract. Exp., 25: 291-304. DOI: 10.1002/spe.4380250305

Vishkin, U., 1985. Parallel pattern matching in string. Inform. Control, 67: 91-113. DOI: 10.1016/S0019-9958(85)80028-0