International Journal of Humanities Education
VOL -13 NO 1 2025 ISSN:2327-0063 | E-ISSN:2327-2457

DESIGN OF THE EFFECTIVE TECHNIQUE TO IMPROVE MEMORY AND TIME CONSTRAINTS FOR SEQUENCE ALIGNMENT

Manpreet Singh
Assistant Professor, Department of Information Technology, Guru Nanak Dev Engineering College, Ludhiana. mpreet@gndec.ac.in

Pankaj Bhambri
Assistant Professor, Department of Information Technology, Guru Nanak Dev Engineering College, Ludhiana. pkbhambri@gmail.com

Shyam Lal
Assistant Professor, Department of Mathematics, Akal Degree College, Sangrur. Shyam.jangra1987@gmail.com

Yashwant Singh
Assistant Professor, Department of Computer Science, Government College, Hisar. yssangwan@gmail.com

Mehzabeen Kaur
Assistant Professor, Department of Computer Science & Engineering, Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib. mehzabeenkaur93@gmail.com

Jagvir Singh
Software Developer, Eurotics, Ludhiana, Punjab, India. jag22@gmail.com

Abstract— Scientists in the medical industry have an enormous amount of information at their disposal because to DNA sequence alignment. DNA alignment, which involves massive effort, is needed to acquire an exact match. It is feasible to generate a phylogenetic tree by sequence alignment to use for phylogenetic analysis. DNA sequence alignment may use a variety of algorithms, depending on a sequence's hardware and software needs. The types of sequencing methods available, including their benefits and limitations, differ based on their use case. The MSA package of R program is implemented in the present work. The sequence alignment is performed with the help of the Clustal-W algorithm. This paper uses three groups of varied nucleotide length. Each sequence represents a distinct species. To keep memory and time needs as low as possible, this work's goal is to do as much as we can. Matching the sequence via MSA takes memory and time, which can be computed using the R software's in-built algorithms. This approach is reviewed and benchmarked based on memory and time, providing a major result in 10.2 seconds while utilizing 75.5 KB of storage and finishing the alignment operation.

Index Terms—Computational Biology, Data Systems, Phylogenetic Analysis, and Sequence Alignment

I. SEQUENCE ALIGNMENT

One of the most essential procedures in which two or more DNA sequences are analyzed to locate different, matching sections between them is DNA sequence alignment. A variety of sequence alignment strategies exist, with the major distinction being local vs. global alignment. [1]. To infer a mutation or common area, alignment of sequences is carried out. A scientist working in the biological field would be well served by understanding the phrase sequence alignment. Sequencing yields plenty of data. Another critical use is showing if two sequences have a common origin or not [10]. It may be useful in tracking how frequently the two comparison sequences are identical to one another. Sequence alignment programs, such as BLAST and FASTA, are used to analyze sequences so that homologous areas can be found [8]. Wide sequence alignment can indeed be classified into two categories based on the sequence's length, and different alignment strategies are connected with distinct algorithms. The various alignment approaches are therefore explained as follows:

- Global alignment: It is usually related with the algorithm Needleman-Wunsch. The fundamental concept behind this approach is for the entire sequence to be aligned. With other words, in these approaches the alignment is made from one end to the other, so the full sequence is being compared and more gaps can occur in this kind of alignment procedures. Thus the sequences in question are compared through one end to the other in order to discover a similar part [4].

```
Seq 1 -   TAGC-GC-GT
Seq 2 -   TA-CA-CAGT
```

Figure 1. Global Alignment

- Local alignment: It is usually linked to the Smith Walesman algorithm. Local alignment often emphasizes a special sequence portion and compares the sequence portion for best alignment. The sequences are usually sub-strings that are matched and the main purpose of the procedure is to improve and obtain the highest alignment score. There is no limit to start and terminate the alignment from anywhere, and trim from anywhere. This technique can yield the optimum alignment score, however the greatest drawback in this process is the temporal complexity that generally surpasses O (n2) which is obvious whenever the sequence is larger. [4].

```
Seq 1 -   CGATAACGTAT
Seq 2 -   --ATAAAC---
```

Figure 2. Local Alignment

- Pairwise sequence alignment: Aligning of the parallel sequence is the way two DNA or protein structures or biological sequences are aligned. The main purpose of this matching is to locate the same area in two biological sequences that can provide some information over the operational, structural or genetic relations between two sequences. This is basic and straightforward. Alignment can be divided into three parts, known as global alignment, semi-global alignment and local alignment. Pairly alignment can deal with numerous biological concerns and can provide vital prediction and phylogenetic analysis information regarding 3D protein structure. Certain stated scoring algorithms are utilized to match and mismatch match for pairs of sequences. The fundamental idea behind the approach is to optimize the alignment result by aligning the two sequences ideally.

```
Seq 1 - 1 KTSSGNGAEDS    11
          ||||||||||
Seq 2 - 1 KTSSGNGAEDS    11
```

Figure 3. Pairwise Alignment

- Multiple sequence alignment: It is the procedure that aligns more than two sequences. Numerous biological strands can be aligned at simultaneously in this kind of sequence alignment technique. In the phylogenetic analysis, retained domains and preserved promates areas, this kind of sequence alignment approach is highly beneficial. Compared to the pair sequence alignment approach, multiple sequence alignment techniques are more difficult. The gradual alignment technique is implemented in most the multiple sequence algorithms. Scoring criteria that indicate matches and mismatch score are also applicable. In this scenario, the memory and computing time are larger than the number and length or sequences. Multiple sequence alignment is achieved by gradual alignment. The alignment of two pairs is then carried out and the next sequences are matched with the first pair and replaced with single pairs, such that a single alignment with each sequence is obtained.

```
Seq 1 -    PQGGGGWGQ
Seq 2 -    PHGGGWGQ
Seq 3 -    PHGGGWGQ
Seq 4 -    PHGGGWGQ
Seq 5 -    PHGGGWGQ
```

Figure 4. Multiple Sequence Alignment

II. METHODS OF PERFORMING SEQUENCE ALIGNMENT

Three approaches are developed here for sequence alignment:

- Dot matrix or dot plot: This approach has two DNA sequences all over the grid and Dot shows the comparable nucleotides in between two sequences. There are also many unwanted matches after alignment, which can be deleted by entering window size & threshold. The alignment is considered to be diagonal. The opposite diagonals are read as reversal as well as the crossing diagonals as palindromes. The dot in a certain grid position is inserted only if the two elements are comparable in the two DNA sequences. This is a straightforward and easy-to-learn strategy, but it provides no alignment score for the performance alignment procedure.
- Dynamic programming: Dynamic programming is an excellent strategy when biological sequences are aligned. Dynamic programming decomposes big problems in sub-problems and then resolves the sub-problem using recessive approaches to create the optimum solutions for the full problem by optimally resolving the sub-problems. In this method, BLOSUM & PAM matrix are employed. The major purpose is to align the sequences optimally by offering the maximum alignment result by aligning the sequence recurrently. The dynamic programming equation is provided below where F is a dynamic programming matrix & S is a replacement matrix and D is a linear gap.

$$F(i,j) = \max \begin{cases} F(i-1,j-1)+S(X_i+Y_i) \\ F(i-1,j)+d \\ F(i,j-1)+d \end{cases} \quad (1)$$

- Heuristic method: When an inquiry sequence is needed for comparisons to each sequence of the database and offers a maximum-to-minimum alignment result. These strategies are employed when work is performed with a huge database. In this sort of alignment method, the

FASTA and BLAST families are used. This approach defines word length, usually a K-tuple variable, based on the fact that the inquiry sequence is compared to a complete database and the alignment is obtained, which normally results in the homology level decreasing.

III. SEQUENCE ALIGNMENT APPLICATIONS

Sequence Alignment is a critical activity for bioinformatics, which tackles the numerous biological issues and is of considerable relevance in carrying out the various biological activities and is therefore regarded as an essential aspect in the biological field. Some of the sequence alignment applications are discussed below:

- Protein Structure Prediction: Sequence Alignment can assist in predicting protein structure. Alignment and study of the sequence of proteins can assist identify whether the protein's structure is tertiary, primary or 3D.
- Protein Family: It also helps us to recognize the protein family and helps us to assess whether or not a predicted protein is part of a previously inferred family.
- Pattern Identification: One major application is that it can help to pinpoint a certain pattern responsible for certain biological or physical functions.
- Defining Homology: Through sequence alignment, the level of homology in between sequences may be defined and helped to discover the ancestor of a specific species and deduce the common ancestor of numerous species.
- Detect Region Prove to Disease: This can help to find the disease-proof sequence area and can help scientists and clinicians choose the optimum technique and treatment.
- Defect Mutation: The mutation can be recognized between the two sequences, and the effects of the specific mutation on the organism can also be studied.
- DNA Regulatory Elements: The harmonization method also helps to locate the different DNA regulatory elements including such binding sides, terminating sites etc.

IV. FASTA AND BLAST

FASTA is usually a toolset for pair sequence alignment with a specific test-based input file format. These files can contain DNA / protein sequences & FASTA is used to compare these sequences to the current database. It is focused on the word patterns to find the matching words. There is a variable named K tup that specifies the word size. The program's speed and performance determine the size of a word. The K multiple size increases can reduce match hits. The FASTA tools are more sensitive than the BLAST tools because they compare the sequence of the query with each sequence inside the database and produce an optimum alignment. In this search kind, the gaps between sequences are permitted and so the program is slowed down. This particular program is useful for the fewer comparable sequences.

BLAST is a fundamental local alignment searching tool that identifies the similarity between both the query sequence as well as the database. This is an extremely efficient and precise method used to compare biological sequences such as DNA, protein and amino acids. BLAST works on two biological sequences to compare individual residues. Since BLAST is an extremely sensitive tool, it does not allow any break between sequences and is comparatively quick to find the similarity between sequences compared to FASTA and optimum technology.

V. LITERATURE REVIEW

Reference[1] illustrates a new S.A. architecture for the alignment of the DNA sequence. It has been designed to improve the computing speed of the previously created core architecture. The proposed framework was a hardware-based design consisting of Processing Element (PE), arranged as an array that was connected or clubby to the pipeline, combining a systematic array of P.E. Each of the P.E. employed in the architecture may operate the Smith Water algorithm. There are additionally three adders & three comparators in the architecture. The primary objectives of the design of the this model were to speed up the core architecture's computing capability, as it targets the Xilinx Spartan-3E FPGA as well as the observation that architecture has been 1.2X quicker than the reported DNA alignment core based on the S.A. system with a grouping capacity of 241 p.E. which may have a highest clock rate of 98.697 MHZ.

Reference [3] presented a way to address the multiple sequence alignment issue. In the suggested study, the above sequences were first classified into distinct subsets or subgroups, then sequences were individually aligned with the concept of the Center Star method (CSM). Then the separately sequenced group was combined with gradual alignment. The K-mer count was utilized to partition the sequences into separate subgroups by establishing the K-mean algorithm. K-mer defines the sequence of the set of characters and was set to 6 in the proposed work. After using the CSM, the score matrix for each subgroup was developed by aligning all pairings of that subgroup and then defining the center sequence, which maximizes the pair score. The gradual alignment was finally done to attain its final alignment and afterwards UPGMA was employed to build dendrograms and then all alignments were integrated in this dendrogram to calculate the resulting alignment. The performance and precision were obtained by the calculation of both the Sum of Pair Scores (SPS) and the phylogenetic analysis, conducted with the EMBL-EBI Clustalw Phylogeny outline tool. For the study, a specific dataset was employed which is the HIV database that is structured and managed by the National Laboratory of Los Alamos. The reservation shows that the scheme is more effective and accurate and can also generate a tree more efficient and accurate. However, the SP score calculated by the algorithm is considerably lower than that of MUSCLE and CLUSTALW since it focuses solely on certain areas.

Reference [7] outlines the efforts to prove that TCAM is a customizable memory device and already functioning switch network hardware to improve and speed up the issue of two DNA sequences. The proposed concept was physically designed to interact with the currently functioning campus networking devices on one nock. The HPE Aruba 5400Zl networking switch was utilized to implement the model. All amino acids are encoded into a 00 for 2-bit, 01 for T, 10 for G and 11 for C binary code. The Ethernet section has 18 bytes of broader information and 46 bytes of data. The model was shown to match sequence of 8 elements in average 2.4E-8 seconds. These results were highly satisfying and so inspire future research, as it is carried out on a node configuration, so multiple node configuration can also improve the model's performance. However, some system restrictions are also present because the model is limited or primarily focused with TCAMS. Therefore, no other programmed memory device can be used.

Reference[12] presents an extendable array structure consisting of several processing units to address the need for continuous processing of huge amounts of data, which may increase efficiency and also improve performance. The Intel-Altera HARP method is utilized to implement it together with the

close integration of software and hardware. The primary aims of this model were to manage a large number of small readings which would produce a high level of output and address the problem of lengthy random & frequent memory access to retrieve the reference genome. The system consisted of 16-PEs, which work separately & enable the SMEM algorithm run in parallel. A separate PE (PMU) management unit was installed which coordinates with C.P.U and also integrates or allows memory to connect with P.E. In addition, dynamic scheduling & automatic load balancing were developed to increase the output and improve the operation. The observation after using this model showed that when comparing with a single thread of C.P.U, the SMEM algorithm was increased by 4X. In addition, the SMEM seeding stage has been enhanced by 26 per cent compared with 16 C.P.U threads and hence the targets must be expanded by a scalable array-based architecture with a large number of PEs.

Reference[18] demonstrates that a method named Map-Reduce Acceleration was implemented to speed up the sequence alignment process. The approach includes several hardware and is also combined with the algorithm of clustering, which divides and supplies data between several nodes in order to perform parallel processing in an effort to improve accuracy and reduce time requirements. The solution proposed was used using the Hadoop framework. When connected with the Hadoop frame, the system's functionality, flexibility and scalability were optimized and further enhanced by the distributed system architecture in the proposed work. Hadoop also enables the system due to its fault tolerance characteristics. The solution presented leverages HDFS files, and that in turn boosts data access efficiency and optimizes time. The proposed approach likely to be the best algorithm for the genome sequencing only with Hadoop framework.

Reference [19] presents a multiple sequence alignment technique called GL samples. The input sequences of the tool were matched according to the sequence similarity. Whenever the resemblance between both the input sequences is larger, the algorithm fully takes the sequences and the alignment on those sequences were global alignment. However, when the sequence similarity tends to be small or average, the alignment involves local alignment and trims the flank region among them. To test and evaluate the model practically, it was employed for three biological applications, including phylogenetic trees, secondary structure prediction of proteins, and the determination of the individual at high risk of cervical cancer. These results suggest that this technology is important for numerous applications in real life. By testing on three different bases, the algorithm was analyzed and compared to various other algorithms. The assessment demonstrate that the suggested algorithm has a higher scoring and is therefore a very effective and efficient instrument among the other state of the arts and can therefore be regarded as a considerable tool in real life applications.

The reference[19] shows the use of Smith-Waterman alignment for the reduction of the Parallel Azure Map (SW-PAMR) using the cloud computing architecture. The Smith Waterman algorithm and its parallel implantation were followed. This work optimized the different porthole of Hadoop Map. The reduction frame by using a bespoke Azure cloud platform based on Map Reduce. The planned work takes two steps to complete the alignment phase. These steps are the map and reduction phase. Following the map phase is a reduced phase. The two phases are performed simultaneously, which in turn boosts the efficiency of the algorithm & optimally leverages the core of the system. Virtual machine node were utilized to validate the new algorithm and also was integrated with the Azure Cloud platform. The observation part means that the performance of the proposed algorithm confronts the positive direction, since a comparison with the Hadoop-based framework increases the length of

the query sequence. It has been observed that the proposed work reaches the average speed of 19.5 that is much greater than that of the Hadoop-based framework. The proposed work likewise promises to be streamlined in memory.
The copyright transfer document must be signed personally by all authors - no exceptions. You must scan and upload the signed trademark form using the related interface on the site. There are some constraints concerning the copyright file length, therefore please follow the instructions provided in the trademark file upload interface carefully.

VI. PROBLEM FORMULATION

Any living entity consists of the A-adenine, G-guanine, C-cytosine and T-thymine sequence DNA. All the physical & biological properties of the organism and the basis of life are defined in the order of these sequences. Any organism's DNA consists of thousands of genes that are tiny portions of DNA at a certain position and have specific order. The big problem is to develop an effective approach for memory and time restrictions in order to align the various DNA sequence. Thus, by matching the sequence, several difficulties relating to bioinformatics can be addressed or solved. An effective and precise method for alignment of the DNA sequence can help to discover the similarity region between the sequences and therefore enable us to predict and represent our common ancestor. As today, enormous amounts of DNA sequencing are being produced; function & role of every gene of every living organism can be studied. The alignment of the DNA sequence also enables us to determine the mutation in any organism's DNA/ gene sequence leading to the development of new spices. Another important task to achieve through the alignment of the DNA sequence is to help to identify or detect the gene sequence that is responsible for any type of disease or gene prone to any disease, thereby helping the medical expert to carry out treatment and other desired actions and precautions.

VII. OBJECTIVES

The following are the objectives for the completion of the proposed work:

- To design an effective DNA sequence alignment technology.
- To improve sequence alignment problem memory and temporal restrictions.
- Comparison with other state-of-the-art procedures.

VIII. METHODOLOGY

The methodology for the work described consists of building an algorithm that optimizes memory and time limits for the sequence alignment procedure. For the design of the desired model R software is employed, because R program offers a strong and interactive solution to numerous bioinformatics challenges. R's computer platform offers the solution and analysis of numerous problems in bioinformatics and a large range of computing solutions for various parts of data science, data mining, statics, machine learning and other high-end graphical application. R software provides a flexible and versatile open source environment. In addition, R contains many packages needed to perform different types of computing. It also develops the new package through the creation of the open source. MSA is a multi-sequence alignment package. MSA itself is a mixture of several functions for the alignment of multiple sequences. For multiple sequence alignment the MSA package includes Clustalw, Clustal Omega and Muscle Algorithm. In the current work, MSA is employed using the default Clustalw

function. Clustalw is among the various sequence algorithms for measurement and weights that work on a new position-specific method. The algorithm Clustalw works in three steps. In the first phase, the algorithm uses dynamic programming to achieve pair-size alignment. Then, after that, a guide tree is built using the next strategy, and then the numerous sequence alignments are eventually carried out using the progressive alignment approach with the middle-phase guide tree. The following is the number of stages utilized in the methodology:

- Collecting and preprocessing data: First of all, data used as an input into the system as a DNA sequence is obtained from the NCBI, an integral part of the United States National Library of Medicine (NLM). More than two sequences are required to perform the multiple sequence alignment. The three sequences required have been collected, Homo sapiens, Musculus a house mouse & African green monkey. The length of the sequences obtained is 23403, 2875 and 227 characters. NC 000019.10, BC085096.1 and K01787.1 are the identifier of the gathered sequences. All three sequences have been collected in a distinct FASTA File format. Now all three sequences need to be merged or placed in the same FASTA File format for the multiple sequence alignment. All of the sequences were therefore compiled in a FASTA format entitled all FASTA sequence with a notepad. This file was loaded into R program and the sequences obtained were examined from several viewpoints. The sequence length and its GC content & description were evaluated.
- Applying MSA algorithm: The dataset is available for multiple DNA sequence alignment after the collection and preprocessing of the DNA sequences. A R package named MSA is utilized for the multiple sequence alignment to do the multiple sequence alignment of all three sequences. The MSA software employs the standard Clustalw technique to align the many sequences with a weighted approach and new position specific score. The package picked fulfills the job in three parts. In the first portion, the function continues with the pairing of the sequences with K tuple, a totally dynamic programming approach. The result of this component is a distance score, which is computed by the closeness of each pair of sequences using a matrix. The following section of the function has been used to build a guide tree via referencing to the distance scores produced in the first portion by use of the neighboring joining method. The last and last stage of a MSA function aligns progressively with the reference tree to yield the resulting multi-sequence alignment.
- Pairwise alignment: The K tuple approach is used to determine all potential alignment pairings of sequence which usually locate all possible substrings now overlapping or not replicated in the query string of length K and also for DNA sequences the value of K often ranges from 2-4 in length. By thus doing, all feasible pairs are aligned and a score is calculated by dividing the total matches by sum of all pairs of 2 aligned sequences. The penalties are then lowered from the similarity score, then divided into 100 and obtained from 1, which yields the lot of differences per site, are reduced. Then the two sequences for the matrix plot are used, in which each matching pair is displayed as a dot. It then determines the diagonals which match the most and aligns the window with the help of the K tuple match in alignment to obtain the maximum alignment score.
- Guide tree construction and progressive alignment: After pair alignment, build the next stage of the algorithm a guide tree to aid the final alignment of several sequences. A Tree Neighbor

Guide Joining Method is employed. This approach uses the similarity score of a preceding phase acquired by K-tuple and converts it into a distance matrix in which every node pair is segregated from those other nodes using average differential value. When all nodes are changed and network modules are split via a single reference or branch, the guidebook process is completed. The gradual alignment is now done with the tree created. The method passes through the tree from of the lower tip and progresses to the root to achieve pair alignment. A UPGMA method is used to calculate the distance of two species. In the above equation d, the distance between species and the length Di,j between Ci and Cj are shown by:

$$D_{i,j} = \frac{1}{n_i + n_j} \sum_{p \in C_i} \sum_{p \in C_j} d(p,q) \quad (2)$$

In the new I iteration process, j is selected that has the shortest distance and therefore is replaced by removing the old row & column in D. New distance can be calculated and replaced by the formula j:

$$D_{(i,j),k} = \left(\frac{n_i}{n_i + n_j}\right) D_{i,k} + \left(\frac{n_i}{n_i + n_j}\right) D_{j,k} \quad (3)$$

- Calculating memory: After the alignment procedure is performed the next duty is to assess the quantity of memory used for the alignment process by the algorithm. To this end, a built-in R software function is utilized to calculate the memory. This process utilizes the Alignment function as such an input and returns the memory utilized by any process to finish its work. The memory return unit by function is generally byte by default that allows the output unit to be manipulated in a desired manner.
- Calculating time: As indicated in the time limit targets, the effectiveness of the algorithm is also a significant element. Therefore, after examining memory needs, the next objective is to compute and analyze the time needed by the method for the sequence alignment task. The approach is used to calculate the time required to match the sequences in the constructed function of R program. The function name is system.time (). This function considers the object as an argument and estimates the time the object takes to fulfill its task. This function in fact calls a function called proc.time(), which calculates the time spent by every process to accomplish its task and returns the CPU time to a user to determine the time for running any function in R program.

IX. RESULT AND DISCUSSION

This section of the paper covers the result of the multiple sequence algorithm produced. The mentioned R algorithm software is used to develop. The primary goal or focus throughout the development of this method was memory and time restrictions. The work is done to build an efficient sequence algorithm to meet the memory and time restrictions or to optimize the time and memory demand limits. The DNA input sequences are obtained from the respiratory NCBI and combined in a single FASTA file. The MSA package is used in R to achieve the goals. As MSA is an algorithm collection, which is aligned by several sequences including Clustalw, Clustal Omega and MUSCLE. The Clustalw is used for the process of alignment. The duration of these three DNA sequences is varied, which are 23403, 2875 and 227, the sequences are NC-000019.10, BC085096.1 and K01787.1; and the sequences are collected from Homo sapiens, Mus Musculus and African Green Ape. Following the processing of these sequences, the result implies a considerable memory and time figure, which is nevertheless optimal in comparison with other artistic conditions. The method only completes in 10.2 seconds the matching of all three sequences. The memory used to align the algorithm to various sequences is 77,244 bytes, which makes up over 75,5 KB.

TABLE I. COMPARISON TABLE

Method	Time	Memory	Sequence Length
Proposed method	10.2 seconds	75.5 KB	227 to 23403
Pointing matrix	11.98	2000 KB	64 to 1024

X. CONCLUSION AND FUTURE SCOPE

This work develops numerous sequence alignments that optimize memory and time restrictions. Multiple sequence alignment is the technique of finding or finding the similarity between sequences between more than two. The MSA package is being used to develop the algorithm. In three phases, the algorithm accomplishes the work. It initially aligns pairs and then builds the guideline tree and a gradual alignment is then carried out to complete pair-wise sequences. Very flexible and strong tool in bioinformatics R software was utilized to complete the work and R's MSA packet was used for processing, including a multi-sequence method Clustalw that produces a decent and efficient result. The observation after the technique is implemented is that the memory and time needs are optimized to a large extent. It takes only 10.2 seconds but only 77244 bytes of memory to align the strings of length 23403, 2875 and 277.

The suggested work can be further extended to align a wide range of sequences with the built-in front end and provide the menu and icon-based graphical Interface, enabling the user to extract relevant information without the in-depth knowledge of a technique computer app and other machine-based command line integer.

ACKNOWLEDGMENT

This work is executed for the completion of Dissertation Work of Masters of Technology in Computer Science and Engineering. Facilities available at Guru Nanak Dev Engineering College were used for the submission of Dissertation to Inder Kumar Gujral Punjab Technical University, Kapurthala, Punjab, India. Dr. Manpreet Singh and Dr. Pankaj Bhambri are the supervisors for this research work.

REFERENCES

[1] D. S. Nurdin, M. N. Isa and S. H. Goh, "DNA sequence alignment: A review of hardware accelerators and a new core architecture," 2016 3rd International Conference on Electronic Design (ICED), Phuket, 2016, pp. 264-268.

[2] X. Ye, P. Feng and J. Kang, "A new Boolean logic algorithm for DNA sequence alignment," 2012 5th International Conference on BioMedical Engineering and Informatics, Chongqing, 2012, pp. 941-944.

[3] K. K. Perera and C. T. Wannige, "A hybrid algorithm for multiple DNA sequence alignment," 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer), Negombo, 2016, pp. 323-323.

[4] Y. Li and Z. Ji, "A local alignment of DNA based on parallelized MUMmer algorithm," 2014 10th International Conference on Natural Computation (ICNC), Xiamen, 2014, pp. 401-406.

[5] J. Zhang and C. Yang, "DNA sequence recognition based on the Markov model," 2013 6th International Conference on Biomedical Engineering and Informatics, Hangzhou, 2013, pp. 536-540.

[6] A. M. Hosny, H. A. Shedeed, A. S. Hussein and M. F. Tolba, "An efficient solution for aligning

huge DNA sequences," The 2011 International Conference on Computer Engineering & Systems, Cairo, 2011, pp. 295-300.

[7] D. V. Garro, C. V. Calderón and C. S. Yeung, "Using a programmable network switch TCAM to find the best alignment of two DNA sequences," 2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI), San Jose, 2016, pp. 1-5.

[8] S. S. Ray, S. Ghosh and R. Prasad, "Low-cost hierarchical memory-based pipelined architecture for DNA sequence matching," 2014 Annual IEEE India Conference (INDICON), Pune, 2014, pp. 1-6.

[9] H. A. Shah, L. Hasan and N. Ahmad, "An optimized and low-cost FPGA-based DNA sequence alignment — A step towards personal genomics," 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, 2013, pp. 2696-2699.

[10] H. L. Chen, F. H. Cheng and S. H. Hu, "A reconfigurable embedded system for sequence alignment problem," 2011 International Conference on Machine Learning and Cybernetics, Guilin, 2011, pp. 1345-1351.

[11] H. S. Lopes and G. L. Moritz, "A distributed approach for a multiple sequence alignment algorithm using a parallel virtual machine," 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, 2005, pp. 2843-2846.

[12] S. Chatterjee, R. A. Smrity and M. R. Islam, "Protein structure prediction using chemical reaction optimization," 2016 19th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2016, pp. 321-326.

[13] S. K. Ray, D. Roy and A. R. Shaikh, "DNA Sequence Alignment Engine: An AM-based design," 2011 Annual IEEE India Conference, Hyderabad, 2011, pp. 1-5.

[14] J. Fan and R. Jiao, "New Fast Algorithm on DNA Sequence Alignment," 2009 3rd International Conference on Bioinformatics and Biomedical Engineering, Beijing, 2009, pp. 1-4.

[15] M. Nordin and A. Rahman, "Utilizing MPJ Express Software in Parallel DNA Sequence Alignment," 2009 International Conference on Future Computer and Communication, Kuala Lumpar, 2009, pp. 567-571.

[16] A. Besharati and Mehrdadjalali, "Multiple sequence alignment using biological features classification," 2014 International Congress on Technology, Communication and Knowledge (ICTCK), Mashhad, 2014, pp. 1-5.

[17] A. R. Ekre and R. V. Mante, "Genome sequence alignment tools: A review," 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2016, pp. 677-681.

[18] A. R. Ekre and R. V. Mante, "Hadoop based clustering system for genome sequencing," 2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM), Chennai, 2016, pp. 22-27.

[19] Y. Ye et al., "GLProbs: Aligning Multiple Sequences Adaptively," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 12, no. 1, pp. 67-78, Jan.-Feb. 1 2015.

[20] S. P. Algur and L. I. Sakri, "Parallelized genomic sequencing model: A big data approach for bioinformatics application," 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Davangere, 2015, pp. 69-74.

[21] C. Zhao and S. Sahni, "Cache and energy efficient alignment of very long sequences," 2015 IEEE 5th International Conference on Computational Advances in Bio and Medical Sciences (ICCAB), Miami, FL, 2015, pp. 1-6.
[22] K. Chaichoompu, S. Kittitornkun and S. Tongsima, "MT-ClustalW: multithreading multiple sequence alignment," Proceedings 20th IEEE International Parallel & Distributed Processing Symposium, Rhodes Island, 2006.
[23] S. Rezaei and M. M. Monwar, "Divide-and-Conquer Algorithm for Clustalw-MPI," 2006 Canadian Conference on Electrical and Computer Engineering, Ottawa, Ont., 2006, pp. 717-720.
[24] C. L. Hung, C. Y. Lin, Fu-Che Wu and Yu-Wei Chan, "Efficient parallel UPGMA algorithm based on multiple GPUs," 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Shenzhen, 2016, pp. 870-873.
[25] A. Layeb and A. H. Deneche, "Multiple Sequence Alignment by Immune Artificial System," 2007 IEEE/ACS International Conference on Computer Systems and Applications, Amman, 2007, pp. 336-342.
[26] D. M. German, B. Adams and A. E. Hassan, "The Evolution of the R Software Ecosystem",17th European Conference on Software Maintenance and Reengineering,vol: 2 , pp. 243-252 , 2013.
[27] C. Ramirez, M. Nagappan and M. Mirakhorli, "Studying the impact of evolution in R libraries on software engineering research", IEEE 1st International Workshop on Software Analytics (SWAN), vol:1, pp. 29-30, 2015.
[28] Paul Torfs & Claudia Brauer, “A (very) short introduction to R”, Wageningen University, The Netherlands, Hydrology and Quantitative Water Management Group, 2014.
[29] Robert Gentleman, “R Programming for Bioinformatics”, Seattle, Washington, Taylor & Francis Group, 2009.
[30] Uma Makheswari M and Sudarsanam D, “A Review on Bio Informatics for Diabetic Mellitus”, Vol 3, No 6 , pp. 389-395, 2003.
[31] K..Saravananathan, T.Velmurugan, ”Impact of Classification Algorithms in Diabetes Data: A Survey”, The 3rd International Conference on Small & Medium Business, vol. 4, pp. 271-275, 2016.
[32] V Chandra Sekhar, “Identification of differentially expressed genes for diabetes with parental history vs healthy using Microarray data analysis”, 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), vol.4, pp. 496-500, 2010.
[33] Lekha S. and Suchetha M.,”Non- Invasive Diabetes Detection and Classification Using Breath Analysis”, IEEE ICCSP conference, vol. 4, pp. 955-958, 2016.
[34] Haidar, "The Artificial Pancreas: How Closed-Loop Control Is Revolutionizing Diabetes",IEEE CONTROL SYSTEMS MAGAZINE, vol. 36, no. 5, pp. 28-47, 2016.
[35] D. Chitkara and R. K. Sharma, "Voice based detection of type 2 diabetes mellitus",International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics, vol. 5, pp. 83-87, 2016.
[36] M. Panwar, A. Acharyya, R. A. Shafik and D. Biswas, "K-nearest neighbor based methodology for accurate diagnosis of diabetes mellitus," 2016 Sixth International Symposium on

Embedded Computing and System Design (ISED), Patna, India, 2016, pp.132-136.

[37] V. R. Balpande and R. D. Wajgi, "Prediction and severity estimation of diabetes using data mining technique," 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bengaluru, India, 2017, pp. 576-580.

[38] W. Xu, J. Zhang, Q. Zhang and X. Wei, "Risk prediction of type II diabetes based on random forest model," 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, India, 2017, pp.382-386.

[39] S. Joshi and M. Borse, "Detection and Prediction of Diabetes Mellitus Using Back-Propagation Neural Network," 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), Ghaziabad, 2016, pp. 110-113.

[40] S. Khanna and S. Agarwal, "An Integrated Approach towards the Prediction of Likelihood of Diabetes," 2013 International Conference on Machine Intelligence and Research Advancement, Katra, 2013, pp. 294-298.

[41] K. Yan and D. Zhang, "A novel breath analysis system for diabetes diagnosis," 2012 International Conference on Computerized Healthcare (ICCH), Hong Kong, 2012, pp. 166-170.

[42] A. Franco and E. León, "Meta-classifier for Type 2 Diabetes Mellitus comorbidities in Colombia," 2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013), Lisbon, 2013, pp. 627-631.

[43] M. NirmalaDevi, S. A. alias Balamurugan and U. V. Swathi, "An amalgam KNN to predict diabetes mellitus," 2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), Tirunelveli, 2013, pp. 691-695.

[44] J. A. Oliveira, G. Minas, M. Correia-Neves, J. Mariz, C. Capela and N. Sousa, "Point-of-Care Testing device for analysis of Diabetes Mellitus," 2013 IEEE 3rd Portuguese Meeting in Bioengineering (ENBENG), Braga, 2013, pp. 1-6.

[45] J. Liu et al., "Caveolae image analysis for pathogen diabetes," 2012 5th International Conference on BioMedical Engineering and Informatics, Chongqing, 2012, pp. 247-250.

[46] S. Rahaman, "Diabetes diagnosis decision support system based on symptoms, signs and risk factor using special computational algorithm by rule base," 2012 15th International Conference on Computer and Information Technology (ICCIT), Chittagong, 2012, pp. 65-71.

[47] S. Siyang, C. Wongchoosuk and T. Kerdcharoen, "Diabetes diagnosis by direct measurement from urine odor using electronic nose," The 5th 2012 Biomedical Engineering International Conference, Ubon Ratchathani, 2012, pp. 1-4.

[48] Claes Ignell, Magnus Ekelund, Eva Anderberg and Kerstin Berntorp, “Model for individual prediction of diabetes up to 5 years after gestational diabetes mellitus”, Ignell et al. SpringerPlus, vol. 5, pp. 1-11, 2016.

[49] Ionela Iancu, Maria Moţa, E. Iancu, “Method for the Analysing of Blood Glucose Dynamics in Diabetes Mellitus Patients”, IEEE International Conference on Automation, Quality and Testing, Robotics, vol. 1, pp. 60-65, 2008.

[50] Jianchao Han, Juan C. Rodriguze and Mohseni Behesht, “Diabetes Data Analysis and Prediction Model Discovery Using RapidMiner”, Second International Conference on Future

Generation Communication and Networking, vol. 1, pp. 96-99, 2008.

[51] K. Zahirnia, M. Teimouri, R. Rahmani and A. Salaq, "Diagnosis of type 2 diabetes using cost-sensitive learning," 2015 5th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, 2015, pp. 158-163.

[52] A. Anand and D. Shakti, "Prediction of diabetes based on personal lifestyle indicators," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2015, pp. 673-676. doi: 10.1109/NGCT.2015.7375206.

[53] N. Chetty, K. S. Vaisla and N. Patil, "An Improved Method for Disease Prediction Using Fuzzy Approach," 2015 Second International Conference on Advances in Computing and Communication Engineering, Dehradun, 2015, pp. 568-572.

[54] Chopra Sumit, Bhambri Pankaj, and Singh Baljit, “Segmentation of Mammogram Images to find Breast Boundaries”, IJCST, vol. 2, no. 2, pp. 164-167, 2011.

[55] Kaur Pradeep, Bhambri Pankaj, “To Design an Algorithm for Text Watermarking”, The SIJ Transactions on Computer Science Engineering & Its Applications, vol. 3, no. 5, pp. 62-67, 2015.

[56] Bhambri Pankaj, and Gupta O.P. “A Novel Method for the Design of Phylogenetic Tree”, International Journal of IT, Engineering and Applied Sciences Research, vol. 1, no. 1, pp. 24-28, 2012.

[57] Bhambri Pankaj, Gupta O.P. “Design of Distributed PreFetching Protocol in Push-to-Peer Video-on-Demand System”, International Journal of Research in Advent Technology, vol. 1, no. 3, pp. 95-103, 2013.

[58] Bhambri Pankaj, and Gupta O.P. “Dynamic Frequency Allocation Scheme of Mobile Networks using Priority Assignment Technique”, International Journal of Engineering and Technology Innovations, vol. 1, no. 1, pp. 9-12, 2014.

[59] Kaur Harleen, Bhambri Pankaj, “A Prediction Technique in Data Mining for Diabetes Mellitus”, Journal of Management Sciences and Technology, vol. 4, no. 1, 2016.

[60] Bhambri Pankaj, Kaur Pradeep “A Novel Approach for Zero Watermarking for Text Documents”, International Journal of Ethics in Engineering & Management Education, vol. 1, no. 1, pp. 30-33, 2014.

[61] Paika Vishal, Bhambri Pankaj, “Edge Detection-Fuzzy Inference System”, International Journal of Management & Information Technology, vol. 4, no. 1, pp. 148-155, 2013.

[62] Kaur Jasmine, Bhambri Pankaj, and Gupta O.P. “Distance based Phylogenetic Trees with Bootstrapping”, International Journal of Computer Applications, vol. 47, no. 24, pp. 6-10, 2012.

[63] Kaur Jasmine, Bhambri Pankaj, and Gupta O.P. “Analyzing the Phylogenetic Trees with Tree-Building Methods”, Indian Journal of Applied Research, vol. 1, no. 7, pp. 83-85, 2012.