PROGRESSIVE ALIGNMENT USING SHORTEST COMMON SUPERSEQUENCE

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Ankush Garg
(Roll No. 801231004)

Under the supervision of:
Dr. Deepak Garg
Head

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, ***"Progressive alignment using shortest common supersequence"***, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of ***Dr. Deepak Garg*** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Ankush

(Ankush Garg)

801231004

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Deepak Garg)
Head, Computer Science and
Engineering Department
Thapar University
Patiala

Countersigned by

(Dr. Deepak Garg)
Head, Computer Science and
Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

No volume of words is enough to express my gratitude towards my guide **Dr. Deepak Garg**, Head, Department of Computer Science & Engineering, Thapar University, Patiala, who has been very concerned and has aided for all the materials essentials for the preparation of this thesis report. He has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am again thankful to **Dr. Deepak Garg**, Head of Computer Science & Engineering Department for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there at the need of hour and provided with all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

(**Ankush Garg)**

801231004

ABSTRACT

The comparison among sequences is very important task in bioinformatics. Sequence alignment provides the better information about comparison among sequences. Alignment of more than two sequences called multiple sequence alignment. Multiple sequence alignment solves many problems of bioinformatics.

Multiple Sequence Alignment is an NP-hard problem. The complexity of finding the optimal alignment is O (L^N) where L is the length of the longest sequence and N is the number of sequences. Hence the optimal solution is nearly impossible for most of the datasets. Progressive alignment solves MSA in very economic complexity but does not provide accurate solutions because progressive alignment has problem of local maxima. There is a tradeoff between accuracy and complexity. Most of the developers are trying to create or enhance the techniques for better accuracy with lesser time complexity.

ClustalW is used for progressive alignment, and ClustalW2.1 is the latest version released till now. Guide tree is a binary tree that guides the alignment of sequences. Guide tree is generated by distance scores between sequences. Distance score is calculated by the alignment score divided by the length of shorter sequence. In this paper, Shortest Common Supersequence (SCS) is utilized to generate the guide tree for progressive alignment and the output alignment results are checked by BAliBASE benchmarks for accuracy. According to SP and TC scores, progressive alignment using the guide tree generated by SCS is better than the guide tree generated by alignment score. Original ClustalW2.1 is modified by SCS, and modified ClustalW2.1 gives better results than the original tool.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

DNA and protein sequences contain all secret of life in living beings. Bioinformatics provides many tools and techniques to find out secret from DNA and protein sequences. Comparing the sequences is the best way to find out the functional responsibility of a gene. Biological sequences are responsible for inheritance in living beings. Children receive structural and functional characteristics from their parents. Every amino acids in protein or nucleotides in DNA are responsible for characteristics in living beings. Comparison of two sequences can be performed by sequence alignment. Alignment of more than two sequences is called multiple sequence alignment (MSA). Multiple sequence alignment is an NP-hard problem, and can be solved by dynamic programming. Bioinformatics provides many techniques to solve the MSA. Time complexity O (L^N) is required for an optimal solution by dynamic programming.

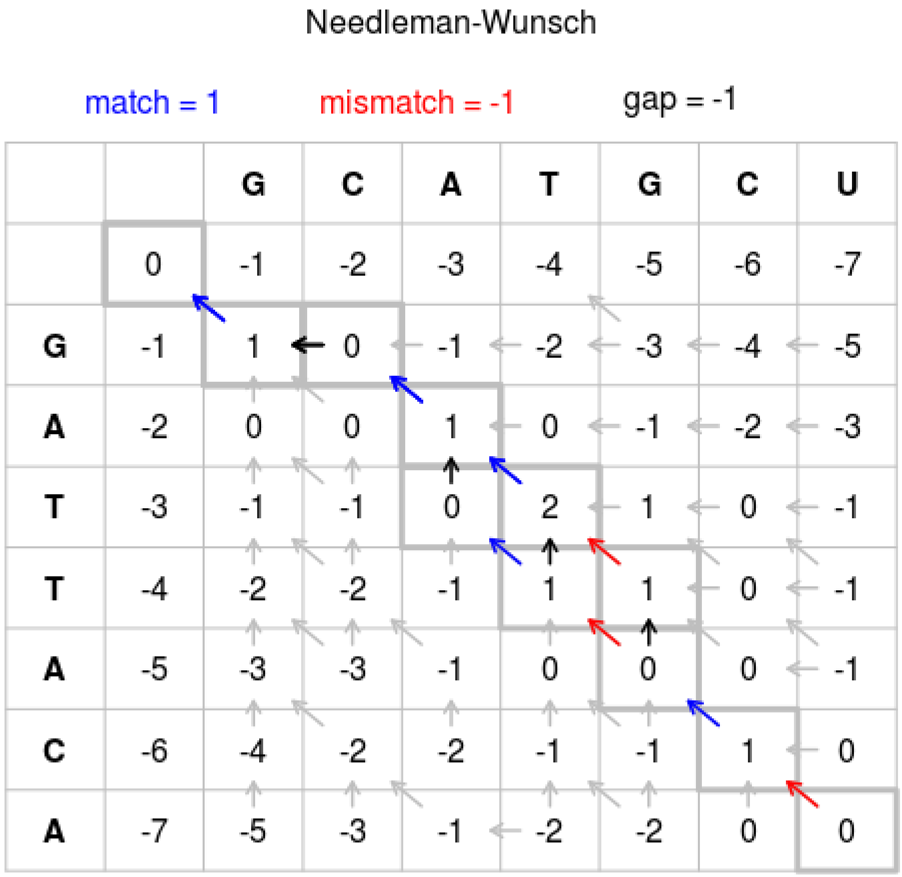
Bioinformatics have many techniques for solving MSA. Some techniques have large time complexity and provide better accuracy, on the other hand, some techniques have economic time complexity but provide a less accurate solution. Progressive alignment is one of the most used technique. Progressive alignment aligns using the step-by-step procedure. ClustalW tool uses the progressive alignment and provides output in many formats according to the requirement. ClustalW is an open source freeware available under GNU public license. Source code of ClustalW is freely available, and modification in ClustalW is very easy due to the modularity.

1.1. Sequence Alignment

Sequence Alignment is the alignment of two sequences using gaps by which alignment score is maximized. Alignment score is calculated on the basis of the number of matches, number of mismatches, number of gaps and number of gap openings. The difference between pattern matching and sequence alignment is gaps that play an important role in alignment score [1].

1.1.1 Global Alignment

Global alignment is alignment considering complete sequence and maximizes the alignment score globally. The Needleman-Wunsch algorithm is the first algorithm for solving global alignment published by Saul B. Needleman and Christian D. Wunsch in 1970 which aligns sequences rapidly. The Needleman-Wunsch algorithm solves algorithm by dynamic programming using matrix shown in Figure 1.1 [2].



		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

Figure 1.1: Matrix generated using Needleman-Wunsch algorithm [3].

1.1.2 Local Alignment

Local alignment considers only some parts of sequences and aligns according to that part. That part of sequences are called regions of similarity. Local alignment aligns the sequences according to a local maximum. Local alignment is very efficient according to the cost but sometimes defining regions of similarity is not an easy task. Local sequence alignment can be solved by Smith–Waterman algorithm [4].

1.2. Multiple Sequence Alignment

Multiple sequence alignment is the alignment of more than two sequences. Multiple sequence alignment (MSA) is very useful in comparison of DNA or protein strands for detecting their evolutionary characteristics. Multiple sequence alignment is an NPcomplete problem and complete for finding optimal solutions is O (L^N) where L is the length of the longest sequence, and N is the total number of sequences. The gaps are used to maximize the alignment scores of sequences. The search for an optimal solution is nearly impossible for a large number of the sequences and long length of sequences. The optimal solution is generated by the dynamic programming. Example of MSA is given below where gap is shown by '–' [5].

```
Q4FK34      MSNLGDVRPVP-HRSKVCRCLFG---PVD
Q9U6R5      ---MAATTAGD-GKRKAARCLFGKPDPEE
P49918      MSDASLRSTST-MERLVARGTFPVLVRT-
Q96TE0      MSNVRVSNGSPSLERMDARQADH---PK-
Q179M8      MS-ARVCNPVALSEIAKLRSPAVVRKP--
Q91603      MAAFHIALQEEMIVASPAALPRLSLGT--
```

1.3. Application of Sequence Alignment

Sequence alignment has vast applications in bioinformatics for research in patterns of sequence of characters, genes and amino acids. Sequence alignment is very powerful technique for comparison between sequences. It gives better information about comparison between sequences then pattern matching. There is few application of sequence alignment discussed below.

1.3.1 Phylogenetic and Taxonomy

Sequence alignment is used for classification of sequences according to the alignment score. Sequences are matched to a database of sequences and sequence are categorized according to the alignment scores with the sequences of database. Sequence alignment also has an ability to find evolutionary relationships accurately between living beings. It is also providing the information about the special characteristics of any living being received from his parents. It also provides the information about the mutation in an organism [6].

1.3.2 Alternative Splicing

Sequence alignment is used in alternative splicing of messenger RNA produced through transcription. Alternative splicing is gene coding for multiple proteins in gene expression. Exons are parts of a gene may merge into mRNA in alternative splicing. Merge operation can be checked by sequence alignment. If two genes are matched from the start to end by sequence alignment then there is very high possibility that both genes evolve from same genes [7].

1.3.3 RNA Editing

In RNA generation through transcription, there is a chance of error in RNA sequence. Either some extra letter is added, or some letter is deleted by mistake. These mistakes some may be very dangerous as single change can convert cells into poisonous cells. These mistakes can be accurately identified by sequence alignment [8].

1.3.4 Genome Assembly

Genome assembly is merging of DNA and RNA sequences for creating new genome. Genome assembly is useful in the treatment of many diseases like cancer. Sequence alignment is used to find a pattern of one genome and try to reconstruct the genome in other organisms [9].

1.3.5 SNP Analysis

SNP stands for single-nucleotide polymorphism. When only single gene is different between two DNA strands then this situation is called SNP. One DNA strand is alleles of other DNA strand. Sequence alignment can easily find the alleles of the gene. For example, flower can be red or white according to change in single genes. This single change in gene can be identified by the sequence alignment [10].

1.3.6 Social Sciences and Business

Sequence alignment has ability of matching sequence of event with the time. Events are like salary change of a person, price change of any product, total sale and purchased. Comparison between these events is very useful in predicting the future event related to corresponding objects. This information is very useful for profit related decisions in daily life [5].

1.3.7Linguistics

There are a large number of languages used in this world, and every language have its own history. One language is generated by other languages. Sequence alignment is useful in identifying the origin of languages with the help of comparison between languages. Most of the language are generated from Latin and origin of language can be checked by using multiple sequence alignment. Sequence alignment has ability to find out similarity between languages [6].

1.4. Problems of Sequence Alignment

Sequence alignment is not easy as its definition and goal for aligning the sequences for maximum alignment score. Increase in numbers of sequences is creating a difficulty with storage, resource requirement, computing time and accuracy of aligned solutions. There are given few major problems of sequence alignment.

1.4.1 Storage Problems

Sequences are increasing with very high speed and their aligned combination also increasing. Sequence alignment requires a large space for storing alignment and large reserve space for future. Storage must be efficient for sequence alignment for facing future problems of scarcity of space. Developers must use some efficient method for storing aligned solution that requires lesser space and enhance the capacity of storage of sequences and aligned solution [11].

1.4.2 Accuracy Problems

Most of the alignment tools are not 100 percent accurate in alignment of sequences because tools with 100 percent accuracy is very slow with time. Accuracy can be enhanced by using different refinement methods. Iterative methods are used for refining the accuracy of sequence alignment. Accuracy depends on the number of iterations and ability of iteration of refinement methods. Accuracy is a major issue for sequence alignment [5].

1.4.3 Time Complexity

Sequence alignment with dynamic programming has very high time complexity and time increases exponentially with the number of sequences and length of sequences for alignment. Developers are trying to reduce the time complexity of alignment tools without affecting the accuracy of tools. Many techniques are used for reducing time complexity like distance measurement through fast Fourier transform [5].

1.5. Motivations and Objectives

Multiple sequence alignment is an interesting topic for research in bioinformatics, and it has vast application area. Developers are still struggling for the tradeoff between accuracy and time complexity of alignment tools. An increasing numbers of sequences enhance the importance of multiple sequence alignment. A little improvement in alignment tools is more valuable than other techniques.

1.5.1 Motivations

Multiple sequence alignment is very powerful technique for comparison among sequences. Multiple sequence alignment provides the much accurate and useful information than pattern matching. There are a vast range of tools for multiple sequence alignment, and some tools have better accuracy, on the other hand, some tools have well time complexity. Comparison of DNA strands provides very useful information about evolutionary relationship among living beings. This information provides treatment of many diseases like cancer. Cancer cells are just like normal cells but some DNA alteration change some properties of cells and convert normal cells to cancer cells. Sequence alignment has ability to identify the altered gene inside the DNA strands. Sequence are increasing rapidly with time, and there is the need of good management and efficient tools for aligning and storing the sequences.

1.5.2 Objectives

The major objective for multiple sequence alignment is to enhance accuracy of existing tools or to develop a new tool which have better accuracy in lesser time complexity than existing tools. Other objectives are to study and compare different types of techniques and tools for alignment of multiple sequence alignment.

1.6. Outline of Thesis

The remaining chapters are summarized into following division.

Chapter 2-is literature survey. This chapter explains the different types of MSA techniques, existing programs and related works.

Chapter 3 - is problem definition. This chapter explains the problem statement.

Chapter 4 - is proposed solution. This chapter explains the proposed solution and explains how it solves the problem of multiple sequence alignment.

Chapter 5 - is experimental result and discussion. This chapter explains a solution of the problem practically and analyzes the results.

Chapter 6 - is conclusion and future scope. This chapter explains conclusion of the complete report and describes future scope for further experiments.

CHAPTER 2
LITERATURE SURVEY

This chapter explains various types of MSA methods, existing tools for alignment, tools for analyzing the accuracy of alignment tools and related work on multiple sequence alignment. Multiple sequence alignment is a vast topic, and various types of methods have been proposed for optimal alignment. Literature survey focuses on similar and related solution of MSA problems.

2.1. Different Types of MSA

There are different types of multiple sequence alignment techniques for different types of sequences with different characteristics and drawbacks. Some methods are useful in accuracy for certain types of sequences and some are useful for all type of sequences.

2.1.1 Progressive Alignment

Progressive Alignment is an economic technique used to find a solution of MSA. Progressive alignment is a heuristic search approach which aligns sequences in a pairwise manner. Alignment score of all pairs of two sequences is calculated and stored in the matrix. That matrix is upper half triangular part of entire N X N square matrix. After generations of triangular matrix, a guide tree is produced with the help of triangular matrix and guide tree is generated in such a way that pair with better alignment score is aligned earlier. Progressive alignment method is not an optimal method according to accuracy, but it provides solutions very economically. If there are extra gaps in the initial alignment then whole alignment process suffered from that alignment and once a gap is inserted, it will never deleted in entire alignment [10].

2.1.2 Iterative Methods

Complete alignment process is repeated many times with some changes in the guide tree and the best solution is generated by iteration of whole alignment process. There are a large number of methods for the iterative alignment and accuracy of alignment depends on the iteration techniques and number of iterations. Accuracy of alignment using iterative method is directly proportional to time complexity [12].

2.1.3 Hidden Markov Models

A hidden Markov model (HMM) is a model in which present state depend on prior states (prior states is out of sight). In first order HMM, only last state is considered for the present state. HMM can be used to identify the genes for genome sequencing. Second ordered HMM is used for initial sequencing. States are visible in simple Markov model. HMM contains hidden states, and probability is used for value of hidden states. In HMM, probability is used for aligning the sequences. Frequency distribution is used for insertion and deletion. Insertion and deletion are the states of the model. HMM works on all possible combination and finds the precise alignment. In profile HMM sequence alignment, the sequences are searched in an extensive library of profiles for enormous scale sequences comparison. Similar sequences are aligned earlier in progressive alignment. Similarity of sequences is scored according to matching profiles of database [20, 21, 22].

2.1.4 Consistency Based Techniques

The larger number of sequences is a significant problem. There is the need of scalability for handling larger numbers of sequences than capacity of existing tools. It uses both global and local alignment methods to improve the accuracy and gives a simple and flexible solution. T-COFFEE and PROBCONS use the consistency based technique. Consistency based tools use homology with other sequences which refine the quality of pairwise alignment [13].

2.1.5 Template Based Techniques

Template based techniques use the information about protein and DNA from a template and aligns according to them. The template may be any structural information or any profile. Template is added to the sequences and then the sequences are aligned. In this technique, sequences are checked for a similar template in the existing database. In some tools, users can select a template manually and can also choose a method for alignment of sequence [16].

2.1.6 Genetic Algorithms

Genetic algorithms are used to optimize the multiple sequence alignment for better efficiency and accuracy. All sequences are divided into fragments and arranged many times with gaps for generating an optimal solution. Genetic algorithm can also be used with progressive alignment which is called GAPAM. In this method, the initial generation generated randomly then child generation is generated by genetic operators. New generation is a mixture of child and parent generation and repeats this until next few generation are same as previous [17].

2.1.7 Simulated Annealing

Simulated annealing is refining method to find a region of an optimal solution. It maximizes the sum of pairs function by using a temperature factor which is used to find speed of going towards the optimal solution. MSASA uses this approach, and this approach can also be used with the genetic algorithm and other techniques for an optimal solution [18].

2.1.8 Phylogeny-Aware Methods

The main objective of all MSA tools is to minimize the number of gaps used in alignment by which gap penalty is minimized. PRANK is software package released in 2008 which improves the alignment. Time complexity of PRANK is more than other common MSA tools like ClustalW and T-COFFEE [19].

2.1.9 Motif Finding

Motif is pattern of nucleotides or amino acids which has some specific structural or functional characteristics. In motif finding sequence alignment, first of all sequences are aligned globally and select the most common area. This common area is motif of sequences and sequences are aligned according to motif [20].

2.2. Existing Programs

There are so many tools available for MSA. Some tools are good in accuracy, and some are good in time and space complexity. There is a tradeoff between accuracy and complexity. There are some common tools are explained briefly.

2.2.1 ClustalW

ClustalW is MSA tool which uses the progressive alignment. ClustalW generates the triangular matrix of the alignment scores between all pairs of sequences. Cell values of i^{th} row and j^{th} column is the alignment score of i^{th} sequence and j^{th} sequence. In ClustalW 2.1 alignment score is divided by length of bigger sequence and resulting values are filled in triangular matrix. Guide tree is generated by distance matrix on the basis of cell values in triangular matrix. Pairwise alignment is performed according to the guide tree. ClustalW tool also have ability to align according to user define guide tree. ClustalW tool can generate output in many formats and it also accept sequences in many formats [1].

2.2.2 MAFFT

MAFFT is an MSA tool which reduces the time complexity because this tool uses the fast Fourier transform. MAFFT finds out regions of similarity quickly and gives results in very less time complexity, and it also finds out the score of alignment very quickly. MAFFT is many times faster than T-Coffee. ClustalW uses some methods that grow time complexity according to large input, but MAFFT does not use any complicated function for alignment. MAFFT uses progressive and iterative correction method. MAFFT accuracy is nearly equal to T-Coffee. MAFFT converts protein and DNA sequences into volume and polarity. Different amino acids are assigned to different vectors that have two parts: one is volume and second is polarity [14].

2.2.3 Muscle

Muscle is using the distance measure which modified after every iteration. Muscle achieves very high accuracy according to all common benchmark of MSA. It uses the refinement in progressive alignment and achieve high accuracy. It replaces the ClustalW as it gives the better result in less time complexity. Muscle uses the kmer-counting algorithm for enhancing efficiency of alignment by calculating distance score rapidly and provides high accuracy [23].

2.2.4 T-Coffee

T-Coffee is Tree based Consistency tool for alignment that uses both global and local alignment methods to enhance the accuracy and gives a simple and flexible solution. T-Coffee uses information of all sequences unlike simple Progressive alignment which uses information about current sequences that are being aligned. T-Coffee uses the Balance Guide Tree (BGT) as a guide tree which is used for parallelism. Consistency based tools use the Optimized Library (OLM) which is used to identify the relationship between sequences [13].

2.2.5 PROBCONS

It uses the consistency based technique for alignment. It is an open source tool which gives more accuracy than many other commonly used tools. It aligns the sequences in

5 steps. First of all, it compute the probability of every letter for all sequences and define maximum expected accuracy. Third step is the consistency transformation. At last, computation of guide tree and computation of MSA is performed [5].

2.3. BAliBASE

BAliBASE is a benchmark that is used to check and compare the accuracy of tools for MSA. BAliBASE 3.0 is the latest version of BAliBASE and has a more refined test case for checking accuracy of tools. BAliBASE has 6255 sequences for testing the accuracy. BAliBASE has a large number of datasets in the form of TFA format and corresponding XML and MSF format given as reference alignment. MSA tools align the dataset given in TFA format and give output in MSF format. If the output of MSA tools is not in MSF format then output is converted to MSF format by help of other tools. BAliBASE compares the output of MSA tools to corresponding reference alignment [21].

2.3.1 Sum-of-Pairs Score

Sum-of-pair score is number of pairs of nucleotides or amino acids correctly aligned. 2 is added to score if same nucleotides or amino acids are aligned and 1 is added to score if any nucleotides or amino acids is aligned with gaps. The score is divided by maximum possible score.

2.3.2 Column Score

Column score is the total number of correctly aligned column. Hence column score is calculated for all columns and divided by total number of columns. Possible values of sum-of-pair and column scores lie between 0 and 1. Column score is generally less than the sum of pair.

2.4. Related Work

Needleman et al. [26] have proposed a method for finding sequence alignment in 1970. This method can search the similarities between DNA and amino acids sequences. Needleman and Wunsch algorithm uses dynamic programming and gives accurate result for solution of global alignment and all common tools generally use this method as intermediate step for alignment. Space complexity of this algorithms is O (L^1L^2) and time complexity is O (L^1L^2) where L^1 is the length of first sequence and L^2 is length of second sequence.

Smith et al. [26] have introduced Smith–Waterman algorithm for finding local sequence alignment in 1981. Smith–Waterman algorithm find most common subsequence in long sequences by help of dynamic programming. This algorithm find local optima of alignment score for sequences. Most common subsequence is called region of similarity. This algorithms is very efficient in time complexity but sometimes identifying region of similarity is difficult task.

Feng et al. [30] have used the Needleman and Wunsch algorithm for solution of multiple sequence alignment efficiently in 1987. Feng et al. proposed progressive alignment which aligns sequence by pairwise alignment. This method of alignment iteratively align the sequences until complete alignment achieved, but it has drawback that error in initial alignment is propagate to complete alignment. It finds local optima for alignment score and gives result in very economical time complexity.

Higgins et al. [31] described the Clustal package which can find alignment of multiple amino acids and DNA strand by using progressive alignment in 1988. This package is very economic with time and space complexity, and it aligns the sequences by help of a phylogenetic tree which is also called guide tree. Guide tree is a binary tree which is generated through alignment scores. Clustal package has a problem of local maximum and alignment output is not optimal.

Thompson et al. [32] introduced BAliBASE benchmark for measurement of accuracy of difference alignment tools in 1999. BAliBASE contains large number of datasets and their reference alignment for calculating accuracy of accuracy of specific alignment. It also contain the program which compares the alignment with reference alignment and gives accuracy score in sum-of-pair and column scores.

Notredame et al. [33] described a new method for fast and accurate solution of MSA problems called T-Coffee in 2000. T-Coffee improve accuracy but the time complexity is larger than tools of Clustal package. It pre-processes all sequences for pairwise alignment and uses libraries of ClustalW and lalign (tool for local alignment) and combines the both libraries and creates an extended library which uses for progressive alignment of sequences.

Katoh et al. [14] proposed a very fast tool for MSA called MAFFT in 2002. MAFFT uses the fast Fourier transformation for finding of region of similarity. It uses simplified scoring system for calculation of MSA which reduces the time complexity. It converts sequences in polarity and volume and uses both progressive and iterative method for alignment.

MUSCLE [24] is a new tool which gives high accuracy with less time complexity given by Robert C. Edgar in 2004. It uses progressive alignment with a profile function which calculate the distance by kmer-counting. It achieves highest accuracy than other commonly used tools. It can align 5000 sequence of average 350 length in 7 minutes. The guide tree used in this tool is generated by kmer-counting algorithm on the basis of distance between two sequences.

Cooper et al. [34] introduced Application for Browsing Constraints for interactive browsing in 2004. Application for Browsing Constraints is java based software for multiple sequence alignment. It displays sequence similarity and location of genes of MSA results and it also have ability to export data in general formats like plain text and tree format. It uses very less memory on standalone computer and have flexible user interface.

Yonatan et al. [17] introduced Faster Algorithms for multiple sequence alignment in 2006. They have discussed about finding optimal solution in predefined matching segments. Dynamic solution for MSA problem is very time consuming and increases exponentially. They introduced many algorithms for enhancing the dynamic programming. Their algorithm finds optimal solution in lesser time then optimal solution.

Nasser et al. [35] developed a method for matching the sequences by fuzzy logic in 2007. They derived a function for matching the sequence for multiple sequence alignment according to several factors like gap opening and extending gap. MSA using LCS algorithm gives inaccurate as the LCS algorithm does not consider the gap penalties.

Hongwei et al. [15] have presented SASAlign which uses simulated annealing in 2007. SASAlign initially align with best available algorithm instead of randomly generated alignment which save time complexity for refinement. Initial alignment is refined by using the simulated annealing. It searches optimal solution of multiple sequence alignment problems.

Church et al. [36] proposed MSA algorithms for parallel and distributed memory supercomputers in 2011. Alignment of large number of sequences of large length is time consuming and not suitable for standalone computer. Parallel computing is used to enhance the speed of alignment for large scale analysis of sequences. All the sequences is clustered into groups for parallel computing without conflict. All partial solution is align for multiple sequence alignment at last stage.

SuiteMSA [37] provides a visual access for sequence evolution and M.S.A. generated output in 2011. It is java based application which provide GUI for direct comparison of multiple output of different MSA tools. This tool also evaluate the consistency of output and provide access to change in existing alignment.

Macedo et al. [38] introduced DIALIGN-TX (iterative heuristic method) for multiple sequence alignment in 2011.DIALIGN-TX uses the dynamic programming and it aligns by merging the regions which have no gap with high similarity. It have ability to run on parallel strategy. It can run on heterogeneous multicore clusters with multiple allocation policies.

Naznin et al. [39] proposed GAPAM for solution of MSA problems by progressive alignment using the genetic algorithm in 2012. Initial solution is generated by randomly generated guide tree. After that guide tree are shuffled by changing position of sequences. Guide tree is shuffled until next few solution giving same best results as the alignment using previous guide tree.

Ortuno et al. [40] proposed multiple objective approach NSGAII (a complete system to optimize multiple sequence alignments) in 2012. Alignment generated by progressive and consistency based approaches are coded using a novel representation. Best part of multiple solutions of a problem is merged for better alignment with better accuracy.

Matos et al. [17] introduced a compression model for MSA blocks in 2013. MSA datasets use very large space for storage and compression model make up most of datasets. Compression model is based on mixture of context models. DNA datasets are increasing day by day and space used by these datasets should be optimized. New compression model utilizes .72 bits per symbol for multiz46way dataset and it enhance the efficiency of storage resource.

Nguyen et al. [39] introduced a knowledge based method for solution of MSA in 2013. Knowledge based method is a mechanism for identifying solution of MSA with less cost. This method divide the sequences into groups or cluster according to similarity and compare cluster with knowledge database. Partial output alignment and guide tree are generated after clustering of sequences. All partial output alignment is merged together by alignment. Refinement at last improve the accuracy of output.

Nguyen et al. [41] have proposed an approach of MSA visualization through gradient vector flow analysis in 2013. This approach extract and visualize the patterns in alignment data. MSA visualization through gradient vector flow is used for large scale MSA in exploration process. It convert alignment into vector field for GVF analysis to extract patterns.

Mokaddem et al. [42] introduced Motalign in 2013. It is a new progressive alignment algorithm which generates guide tree by using profile base distance. Profile distance is calculated by according to type of sequences. Complexity of Motalign algorithm is O $(N^4 + NL^2)$ which is also the complexity of refinement step.

CHAPTER 3
PROBLEM DEFINITION

There are N numbers of sequences of DNA or protein strands. The problem is to align the sequences to maximize the alignment score. The score can be calculated by following equation (1). If the ith character of the first sequence (S_i) is compared with the jth character of the second sequence (S_j), then a corresponding score can be calculated as the following equation.

$$Score\ (S_i, S_j) = \begin{cases} 2, if\ Si = Sj \\ 0,\ if\ Si \neq Sj\ \&\ Si \neq '\text{-}'\ \&\ Sj \neq '\text{-}' \\ \text{-}1,\ if\ Si \neq Sj\ \&\ Si = '\text{-}'\ or\ Sj = '\text{-}' \end{cases} \quad (1)$$

From the equation (1), the alignment score can be generated by sum of the alignment scores of all columns but one more penalty for gap opening should be considered [10]. Whenever new gaps are opened then penalty of two is deducted from the alignment score. The total alignment score should be maximized. The gaps should be placed carefully because extra gaps can affect the accuracy. In Progressive alignment, the alignment score between all sequences is generated for generating triangle matrix. A guide tree is generated by triangle matrix according to the value of the alignment scores. S. Hosni et al. explain a new approach for generating guide tree by the help of the LCS (longest common subsequence). This approach has the drawback that If there are two sequences of length L_1 and L_2 where L_1 is smaller than L_2 and the sum of all the lengths of their LCMs is X, and the score is 1 – X/L_1. This score depends only on L_1, but it should depend on L_2 or bigger sequence also. For example, the first string is AGAG and second string AGCC than length of the substring is 2 and min length of the string is 4. If the second string is AGCCCC then result will be same. Hence there is a problem because new progressive alignment approach does not depend on larger string.

Traditional progressive alignment approach can be modified if guide tree generated by such a way so that generated alignment have maximum accuracy. There are a large number of method which refine guide tree and align the sequences with better accuracy. These methods use some extra time for refinement and complexity of alignment increases with the number of iterations for generating and checking guide tree for better accuracy. Iterative refinement method increases the computational time by multiple of iteration of traditional progressive alignment. Finding optimal guide tree by using dynamic programming increases the computational time by factorial of the number of sequences. Hence finding optimal guide by dynamic method is not economic according to the time complexity.

Traditional progressive alignment generates the guide tree using the alignment score between the sequences, and this guide tree is not an optimal tree. Guide tree generated by the LCS is also not optimal, but probability of accurate guide is better than guide tree generated by the alignment score. Progressive alignment using LCS does not increase the time complexity of the alignment. Progressive alignment using LCS have better accuracy than traditional progressive alignment, but there are still scope for better accuracy.

CHAPTER 4
PROPOSED SOLUTION AND IMPLEMENTATION

The problem defined in the previous chapter is solved in this chapter with the help of shortest common supersequence. The solution for multiple sequence alignment problem is explained in following sub-sections.

4.1. Proposed Solution

In Progressive alignment, the guide tree is generated by the alignment scores of all sequences. SCS of all pairs of two sequences is calculated for generating a triangular matrix. The value in i^{th} row and j^{th} column is the sum of the lengths of both sequences divided by the length of SCS of ith sequence and jth sequence. If the matrix is divided by diagonal then only one triangle (upper triangle of the matrix) is needed to form a guide tree.

$$M_{ij} = \frac{Li+Lj}{LSij} \qquad (1)$$

M is the triangular matrix with M_{ij} cells value correspond to ith row and jth column. L_i, L_j and LS_{ij} are the length of i^{th} sequence, length of j^{th} sequence and length of SCS of i^{th} and j^{th} sequences respectively.

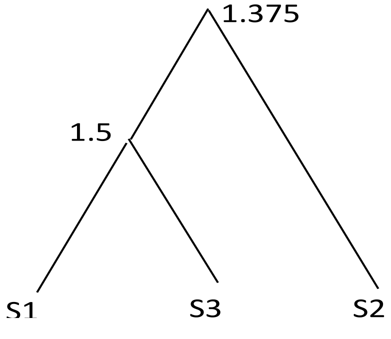


Figure 4.1: Guide tree generated by SCS score.

Suppose S_1 = AGTCGT, S_2 = TCTGA and S_3 = AGCTAC are three sequences to be aligned with progressive alignment using the SCS. First of all, matrix M is generated by the help of SCS algorithm. SCS for S_1 and S_2 is AGTCTGTA, SCS for S_1 and S_3 is AGCTACGT and SCS for S_2 and S_3 is TAGCTGAC. LS_{12}, LS_{13} and LS_{23} are 1.375, 1.500 and 1.375 respectively. The guide tree (shown in Figure 4.1) is guiding to align S_1 and S_3 after that S_2 joined the alignment.

SCS Algorithm

SCS Length Algorithm
Input: sequence$_1$, sequence$_2$, size of sequence$_1$, size of sequence$_2$
Output: Length of SCS

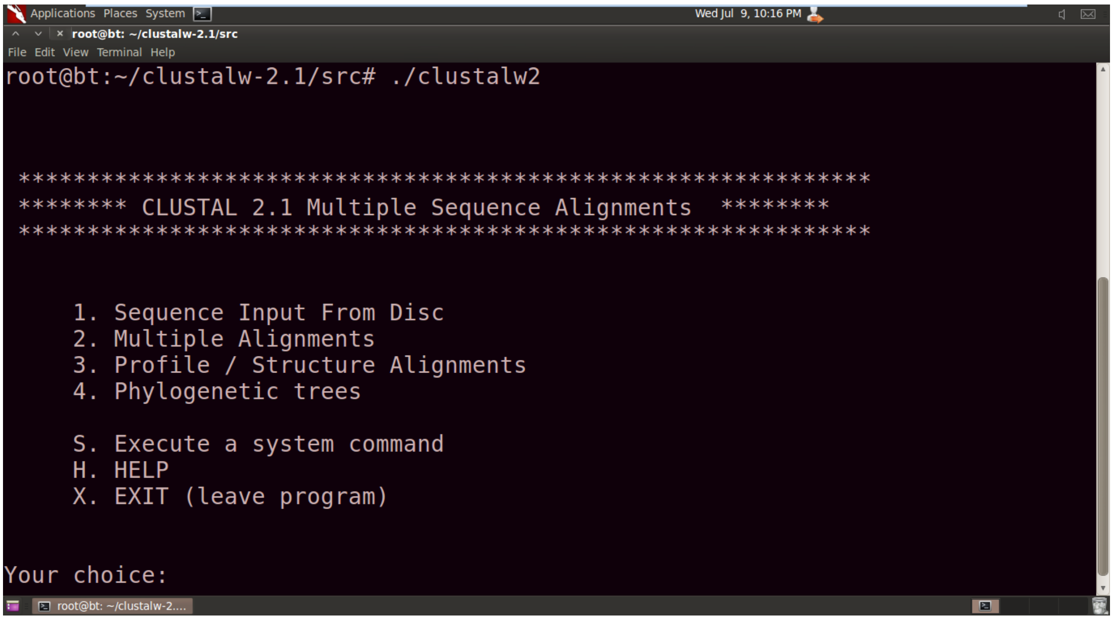
```
procedureSCS_length (sequence₁, sequence₂, L₁, L₂)
    for i← 0 to L₁
        scsarray[L₂][i] ←i
    end for
    for j ← 0 to L₂
        scsarray[j][L1] ← j
    end for
    for i← L₁ to 0
        for j ← L₂ to 0
            if (sequence₁[i] == sequence₂[j])
                scsarray[i][j] ← scsarray[i + 1][j + 1] + 1
            else
              scsarray[i][j] ← min (scsarray[i + 1][j], scsarray[i][j + 1]) + 1
            end if
        end for
    end for
    return scsarray[0][0]
```

SCS length algorithm [43] is given above, and it is used for calculating length of shortest common supersequence of two sequences sequence$_1$ and sequence$_2$. L_1 is the size of first sequence and L_2 is the size of second sequence. This algorithm uses space complexity of O (L_1*L_2) because it uses the scsarray of length (L_1 + 1) * (L_2 + 1).

4.2. Implementation

ClustalW2.1 is widely used tool for progressive alignment, and it gives output in many formats which are easy to use for analysis of sequences of genes. Source code of ClustalW2.1 is modularized and scalable. Modification in this tool is very easy due to modularization property. Guide tree is generated by alignment scores between two sequences. Output of the alignment score is replaced by the shortest common supersequence. Progressive alignment using SCS is achieved by ClustalW2.1 tool, and results are checked by the BAliBASE benchmark for accuracy. Datasets of RV11 and RV12 is utilized for analyzing the accuracy of the modified technique. RV11 contains the divergent sequences and RV12 contains similar sequences.

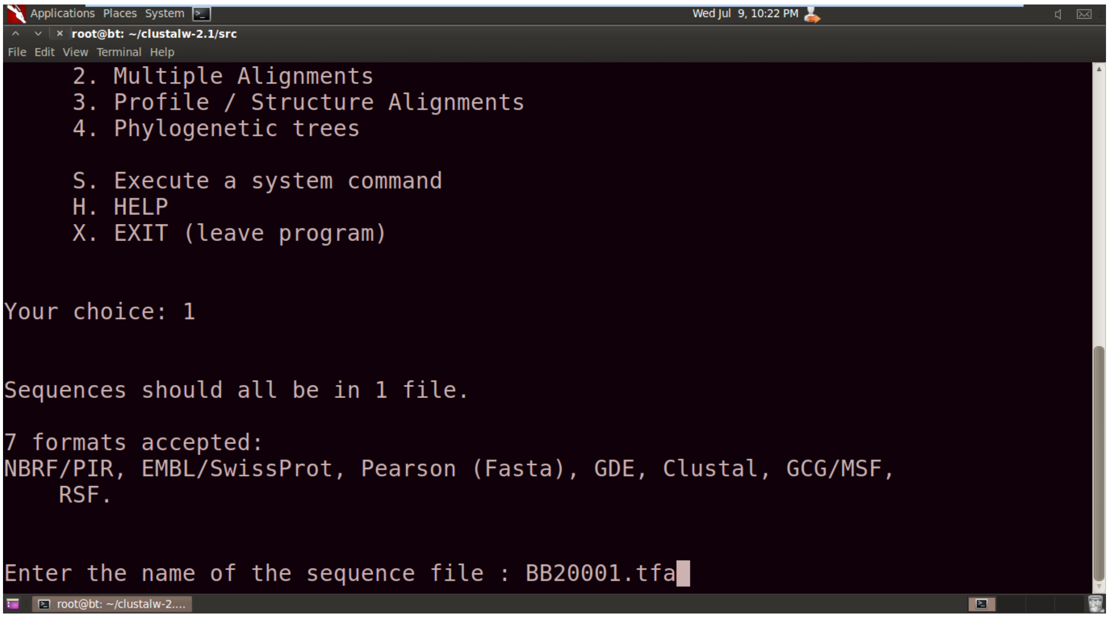
Linux is used as Operating system for the complete experiment. There are large number of tools available for Linux. Most of the tools available for Linux is open source and freeware. Backtrack 5R3 is Linux operating system which contain a large number of already installed tools. Hence Backtrack 5R3 is utilized for installing ClustalW2.1 and BAliBASE. ClustalW2.1 is a GNU licensed product and source code of ClustalW is freely available for changing the code. BAliBASE is benchmark for testing the accuracy of alignment tools. XML parser is required for installing BAliBASE tool on Linux and path of xml parser is used for running the make file.


Applications Places System
Wed Jul 9, 10:16 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```
root@bt:~/clustalw-2.1/src# ./clustalw2



 **************************************************************
 ******** CLUSTAL 2.1 Multiple Sequence Alignments  ********
 **************************************************************


     1. Sequence Input From Disc
     2. Multiple Alignments
     3. Profile / Structure Alignments
     4. Phylogenetic trees

     S. Execute a system command
     H. HELP
     X. EXIT (leave program)


Your choice:
```

Figure 4.2: Homepage for ClustalW2.1.

ClustalW is command line alignment tool in Linux and its source code is modified and compiled for generating executable binary for modified ClustalW tool which uses the SCS. Configure and install make command is used for compiling the source code of ClustalW. Figure 4.2 is showing starting home page for ClustalW, which have many options. Sequence input from disc is used for inserting sequences for alignment in TFA format. Multiple alignments contain multiple options for complete alignment and generating guide tree. Profile/structure alignments option is used for alignment using database of profile. Phylogenetic tree is used for calculating phylogenetic tree which show mutation for all sequences. Executing a system command provides the direct access for code. Help and exit is for assistance and close the tool respectively.


Applications Places System
Wed Jul 9, 10:22 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

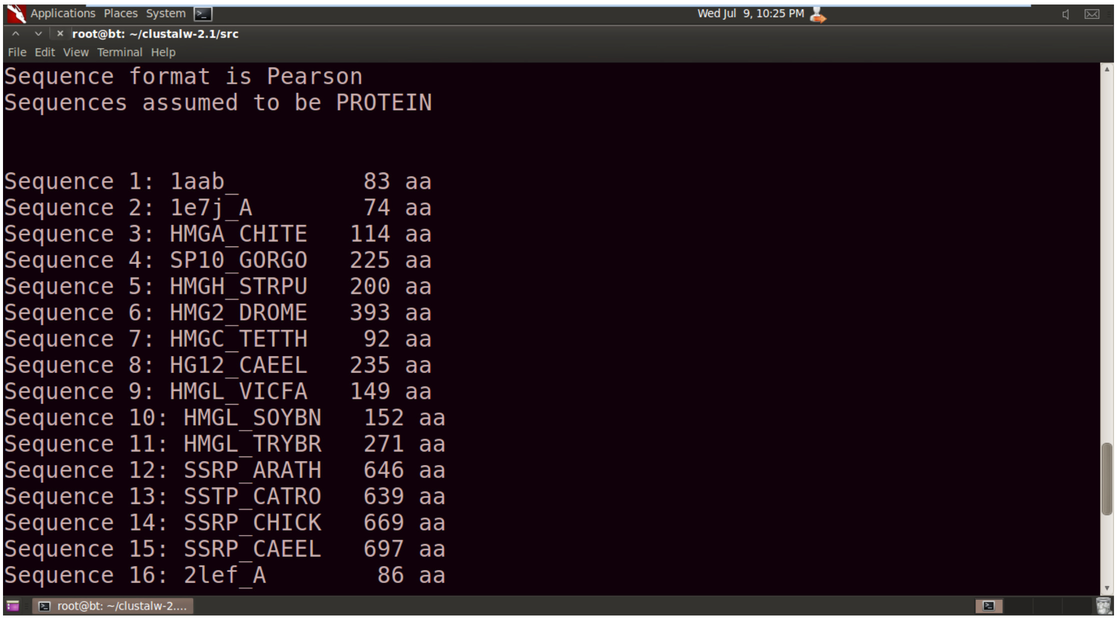
```
     2. Multiple Alignments
     3. Profile / Structure Alignments
     4. Phylogenetic trees

     S. Execute a system command
     H. HELP
     X. EXIT (leave program)


Your choice: 1


Sequences should all be in 1 file.

7 formats accepted:
NBRF/PIR, EMBL/SwissProt, Pearson (Fasta), GDE, Clustal, GCG/MSF,
    RSF.


Enter the name of the sequence file : BB20001.tfa
```

Figure 4.3: Sequence input from disc.

Users provide the name of sequences file after selecting the sequence input from disc shown in Figure 4.3. ClustalW have ability to process the input file in NBRF/PIR, EMBL/SwissProt, Pearson (Fasta), and GDE, Clustal, GCG/MSF and RSF format. All sequences are stored in memory after providing input file. Sequences are stored in memory as integer arrays. Every different amino acids are stored in with different number ranged from 0 to 23.

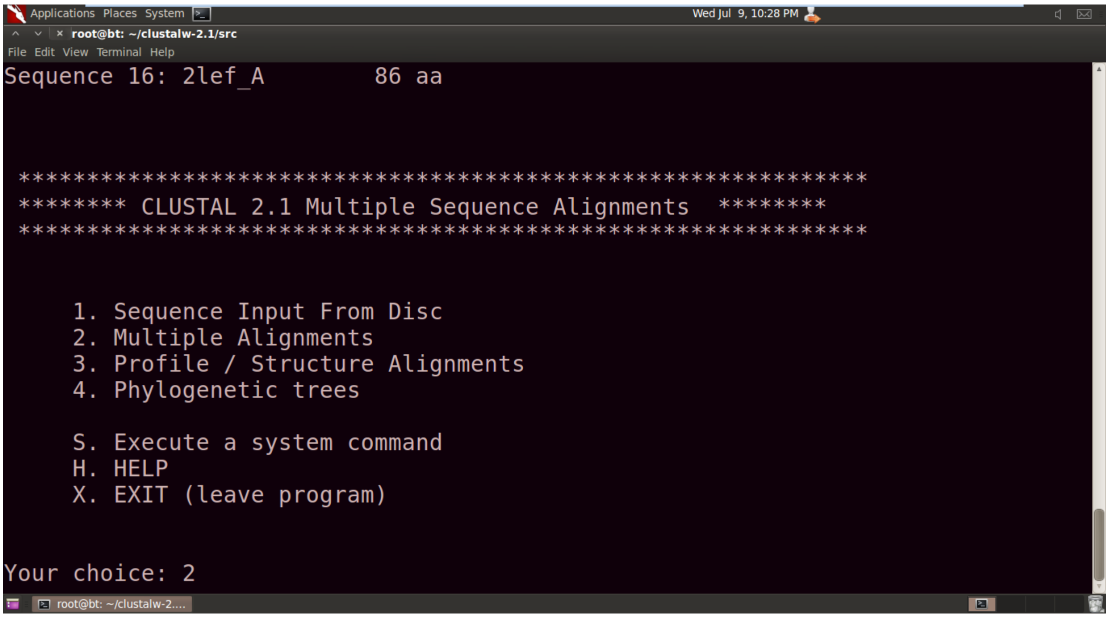
BAliBASE benchmark provides the datasets which contains sequences in TFA format (Pearson format) and corresponding reference alignment in MSF and XML format. BAliBASE also contains software for checking accuracy of output alignment solution. BB20001 datasets which belong to RV20 is used in Figure 4.3.

Figure 4.4 are showing names of all sequences stored in memory for further process. laab, le7j_A and HMGA_CHITE etc. are names of the sequences. Figure 4.4 are also presents the length of sequences. Sequence 1 contains laab_ named sequence, and it contains 83 amino acids. Sequence 2 contains le7j_A named sequence, and it contains 74 amino acids. All sequences are stored in memory in integer form for better time complexity. Integer range 0 - 23 is used for storing the sequences of amino acids in memory of the system. ClustalW uses 2D array of integer for storing all sequences at one place. Rows of array represent the sequences stored in memory. Columns of 2D array represent the characters of sequences. There are 26 characters in English language, but integer range 0-23 used because ‘B’ and ‘O’ is not valid characters.


Applications Places System
Wed Jul 9, 10:25 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```
Sequence format is Pearson
Sequences assumed to be PROTEIN


Sequence 1: 1aab_          83 aa
Sequence 2: 1e7j_A         74 aa
Sequence 3: HMGA_CHITE    114 aa
Sequence 4: SP10_GORGO    225 aa
Sequence 5: HMGH_STRPU    200 aa
Sequence 6: HMG2_DROME    393 aa
Sequence 7: HMGC_TETTH     92 aa
Sequence 8: HG12_CAEEL    235 aa
Sequence 9: HMGL_VICFA    149 aa
Sequence 10: HMGL_SOYBN   152 aa
Sequence 11: HMGL_TRYBR   271 aa
Sequence 12: SSRP_ARATH   646 aa
Sequence 13: SSTP_CATRO   639 aa
Sequence 14: SSRP_CHICK   669 aa
Sequence 15: SSRP_CAEEL   697 aa
Sequence 16: 2lef_A        86 aa
```

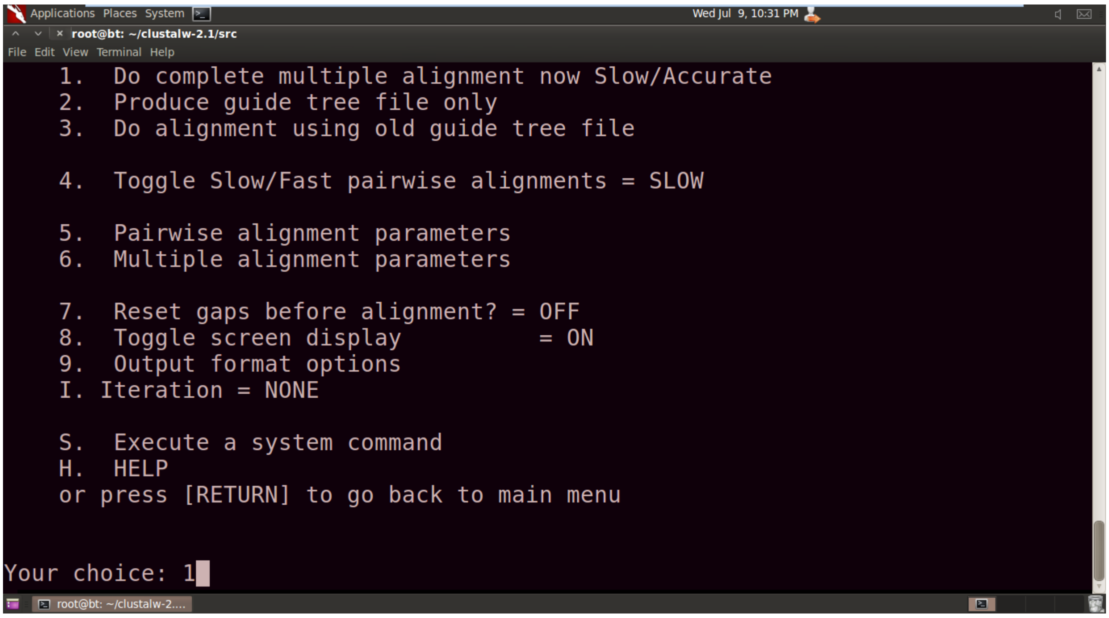
Figure 4.4: Sequences stored in memory


Applications Places System
Wed Jul 9, 10:28 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```
Sequence 16: 2lef_A        86 aa



 **************************************************************
 ******** CLUSTAL 2.1 Multiple Sequence Alignments  ********
 **************************************************************


     1. Sequence Input From Disc
     2. Multiple Alignments
     3. Profile / Structure Alignments
     4. Phylogenetic trees

     S. Execute a system command
     H. HELP
     X. EXIT (leave program)


Your choice: 2
```

Figure 4.5: Multiple alignments.

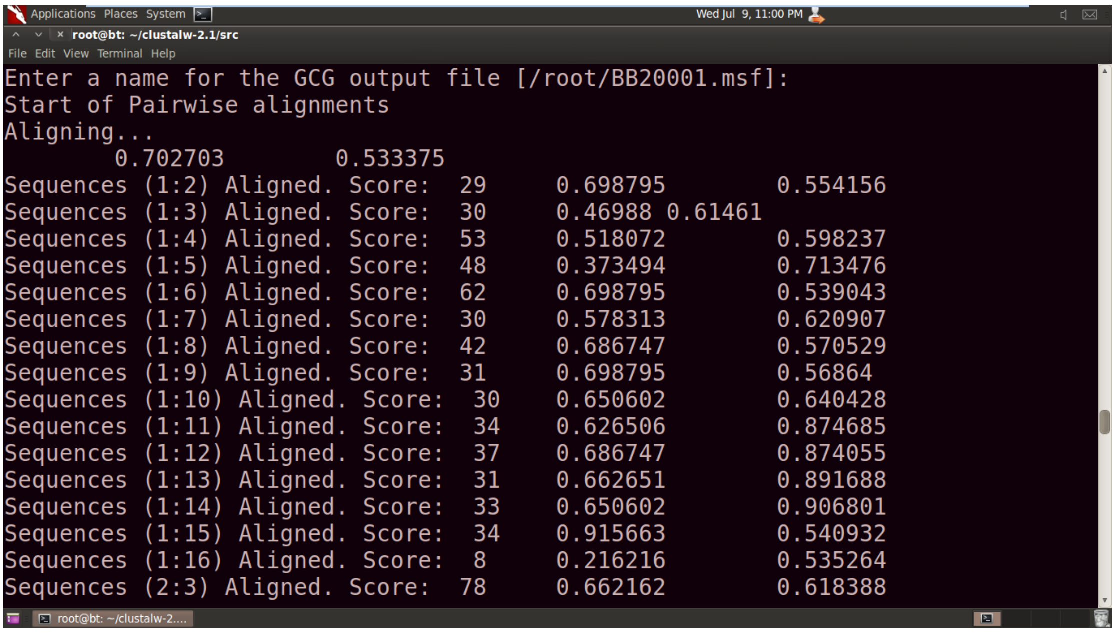
There are many options for user-friendly alignment solution after selecting multiple alignments option. “Do complete multiple alignment now slow/accurate” option generates complete alignment solution, and user can also generate only guide tree by selecting the second option. ClustalW tool also has the ability to generate alignment by providing guide tree for user’s input, and users can also change the format of the output shown in Figure 4.6. ClustalW also has the ability to change modes of alignment slow and fast. Slow alignment mode align the sequences with better accuracy, on the other hand, fast mode aligns the sequence in lesser time.


Applications Places System
Wed Jul 9, 10:31 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```
    1.  Do complete multiple alignment now Slow/Accurate
    2.  Produce guide tree file only
    3.  Do alignment using old guide tree file

    4.  Toggle Slow/Fast pairwise alignments = SLOW

    5.  Pairwise alignment parameters
    6.  Multiple alignment parameters

    7.  Reset gaps before alignment? = OFF
    8.  Toggle screen display          = ON
    9.  Output format options
    I. Iteration = NONE

    S.  Execute a system command
    H.  HELP
    or press [RETURN] to go back to main menu


Your choice: 1
```

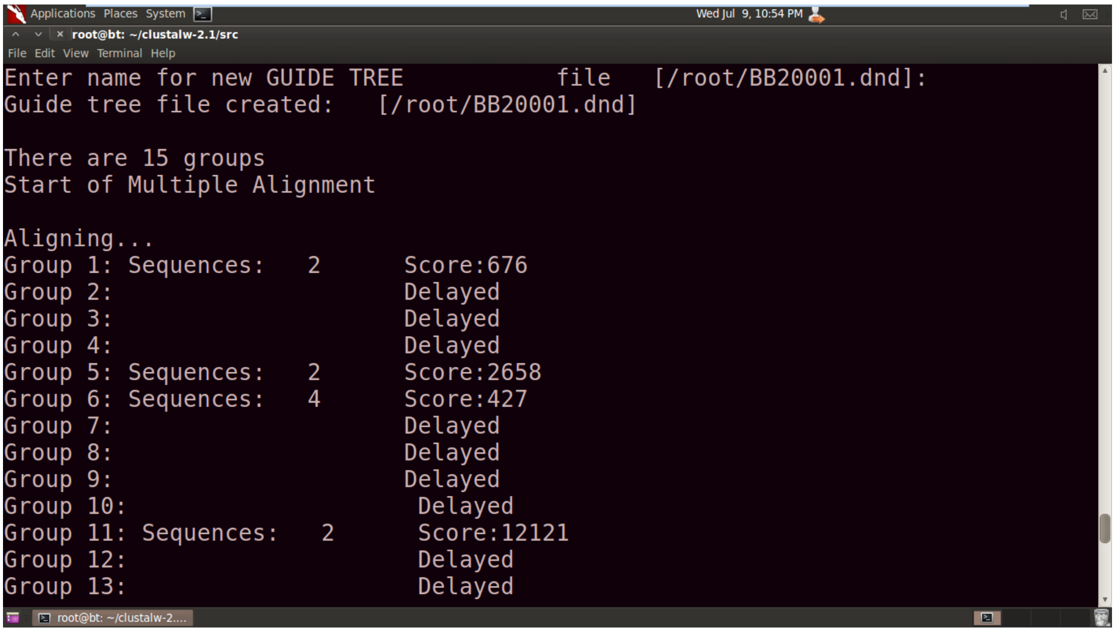
Figure 4.6: Options after multiple alignments.

Modified ClustalW is generating guide tree after selecting complete alignment option in Figure 4.7. It generates the guide tree in DND format for further alignment. Guide tree is generated by help of SCS. By default, guide tree is generated in the directory of the input file and name same as the name of input file but with a different extension. SCS distance score pair of sequences are given in Figure 4.7. SCS distance score for sequence1 and sequence2 is 0.808917. Alignment score and distance score for sequence 1 and sequence 2 is 28 and 0.0.716216.


Applications Places System
Wed Jul 9, 11:00 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```
Enter a name for the GCG output file [/root/BB20001.msf]:
Start of Pairwise alignments
Aligning...
        0.702703        0.533375
Sequences (1:2) Aligned. Score:  29     0.698795        0.554156
Sequences (1:3) Aligned. Score:  30     0.46988 0.61461
Sequences (1:4) Aligned. Score:  53     0.518072        0.598237
Sequences (1:5) Aligned. Score:  48     0.373494        0.713476
Sequences (1:6) Aligned. Score:  62     0.698795        0.539043
Sequences (1:7) Aligned. Score:  30     0.578313        0.620907
Sequences (1:8) Aligned. Score:  42     0.686747        0.570529
Sequences (1:9) Aligned. Score:  31     0.698795        0.56864
Sequences (1:10) Aligned. Score:  30    0.650602        0.640428
Sequences (1:11) Aligned. Score:  34    0.626506        0.874685
Sequences (1:12) Aligned. Score:  37    0.686747        0.874055
Sequences (1:13) Aligned. Score:  31    0.662651        0.891688
Sequences (1:14) Aligned. Score:  33    0.650602        0.906801
Sequences (1:15) Aligned. Score:  34    0.915663        0.540932
Sequences (1:16) Aligned. Score:  8     0.216216        0.535264
Sequences (2:3) Aligned. Score:  78     0.662162        0.618388
```

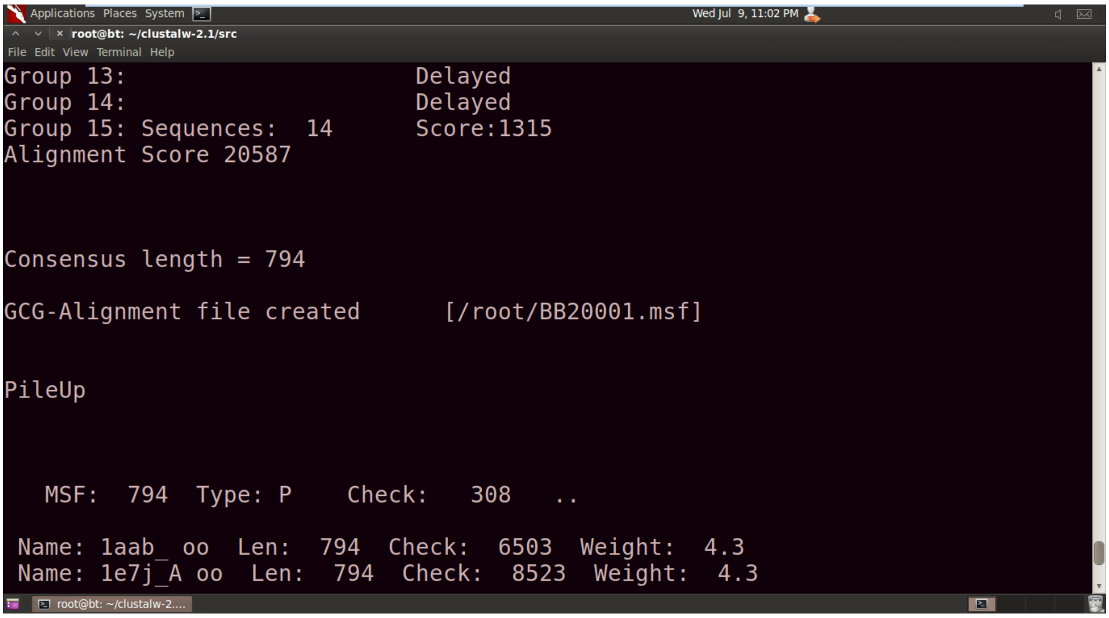
Figure 4.7: Generating pairwise alignment in MSF format.


Applications Places System
Wed Jul 9, 10:54 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```
Enter name for new GUIDE TREE           file   [/root/BB20001.dnd]:
Guide tree file created:   [/root/BB20001.dnd]

There are 15 groups
Start of Multiple Alignment

Aligning...
Group 1: Sequences:   2      Score:676
Group 2:                     Delayed
Group 3:                     Delayed
Group 4:                     Delayed
Group 5: Sequences:   2      Score:2658
Group 6: Sequences:   4      Score:427
Group 7:                     Delayed
Group 8:                     Delayed
Group 9:                     Delayed
Group 10:                     Delayed
Group 11: Sequences:   2      Score:12121
Group 12:                     Delayed
Group 13:                     Delayed
```

Figure 4.8: Generating guide tree in DND format.

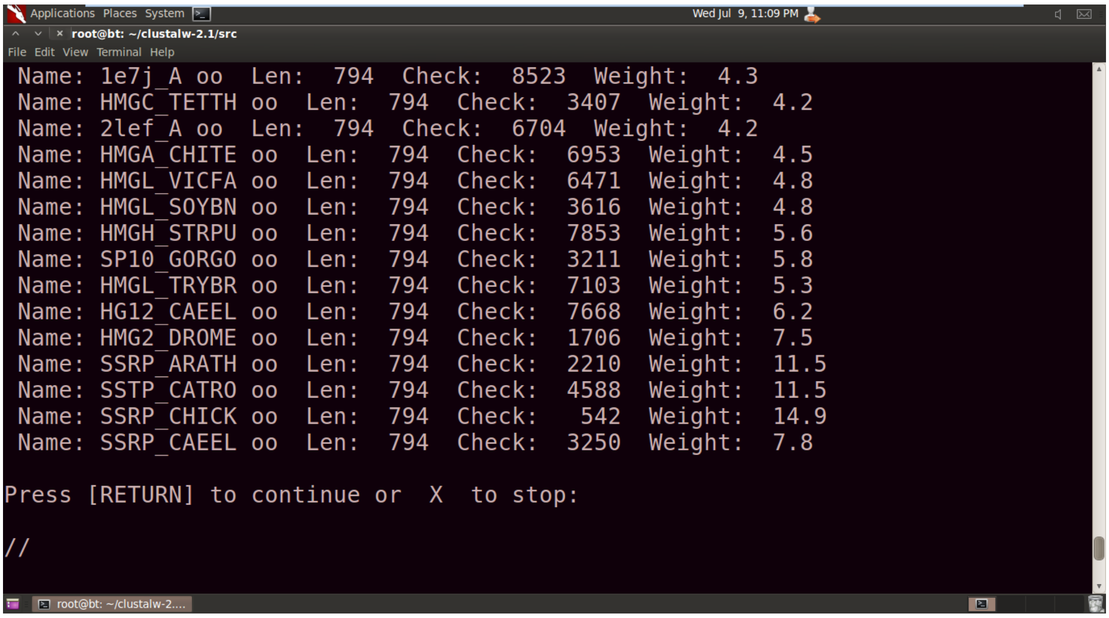
Figure 4.9 is showing alignment score for output alignment generated by modified ClustalW tool. It also provides the information about length of generated output.


Applications Places System
Wed Jul 9, 11:02 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```
Group 13:                     Delayed
Group 14:                     Delayed
Group 15: Sequences:  14      Score:1315
Alignment Score 20587



Consensus length = 794

GCG-Alignment file created      [/root/BB20001.msf]


PileUp



   MSF:  794  Type: P    Check:   308   ..

 Name: 1aab_ oo  Len:  794  Check:  6503  Weight:  4.3
 Name: 1e7j_A oo  Len:  794  Check:  8523  Weight:  4.3
```

Figure 4.9: Alignment score of complete alignment.

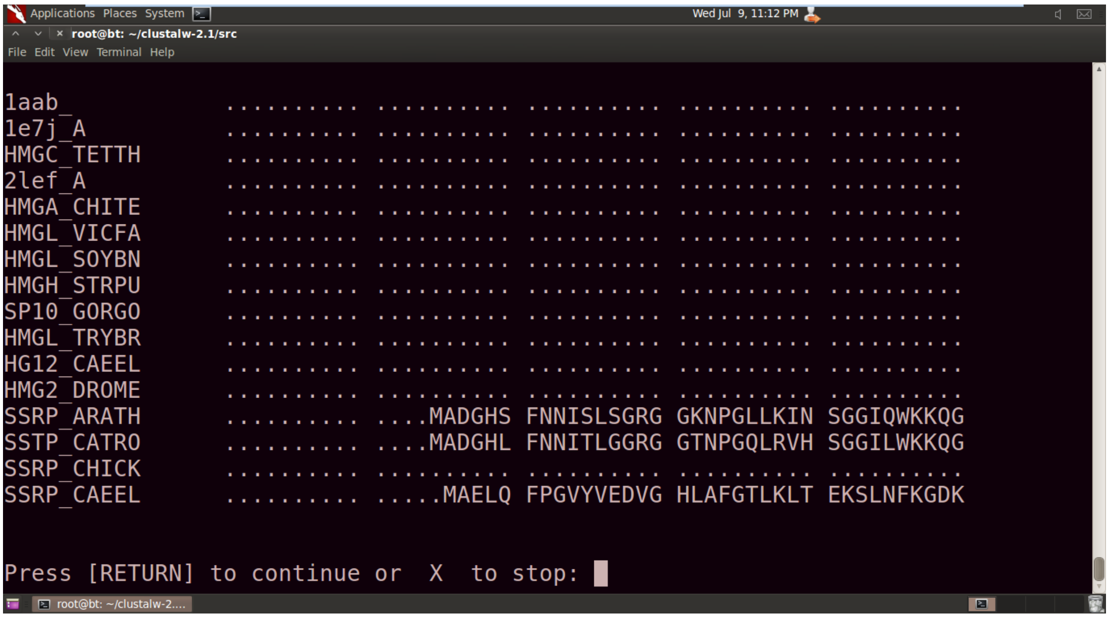
Output generated in MSF format by ClustalW tool is shown in Figure 4.10. Starting lines of MSF format gives the information about all sequences like length, the check score and weight of the sequences. ClustalW gives output alignment in ALN format by default, but user can change the format of the output to GCG/MSF. This format show contribution of sequences in complete alignment.


Applications Places System
Wed Jul 9, 11:09 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```
 Name: 1e7j_A oo  Len:  794  Check:  8523  Weight:  4.3
 Name: HMGC_TETTH oo  Len:  794  Check:  3407  Weight:  4.2
 Name: 2lef_A oo  Len:  794  Check:  6704  Weight:  4.2
 Name: HMGA_CHITE oo  Len:  794  Check:  6953  Weight:  4.5
 Name: HMGL_VICFA oo  Len:  794  Check:  6471  Weight:  4.8
 Name: HMGL_SOYBN oo  Len:  794  Check:  3616  Weight:  4.8
 Name: HMGH_STRPU oo  Len:  794  Check:  7853  Weight:  5.6
 Name: SP10_GORGO oo  Len:  794  Check:  3211  Weight:  5.8
 Name: HMGL_TRYBR oo  Len:  794  Check:  7103  Weight:  5.3
 Name: HG12_CAEEL oo  Len:  794  Check:  7668  Weight:  6.2
 Name: HMG2_DROME oo  Len:  794  Check:  1706  Weight:  7.5
 Name: SSRP_ARATH oo  Len:  794  Check:  2210  Weight:  11.5
 Name: SSTP_CATRO oo  Len:  794  Check:  4588  Weight:  11.5
 Name: SSRP_CHICK oo  Len:  794  Check:   542  Weight:  14.9
 Name: SSRP_CAEEL oo  Len:  794  Check:  3250  Weight:  7.8

Press [RETURN] to continue or  X  to stop:

//
```

Figure 4.10: Output alignment shown in MSF format.

Generated output alignment for BB20001 is shown in Figure 4.11. Sequences are shown with gaps which are represented by "." in MSF format. Output is already stored in disc and command line output representation can be stopped by pressing "X".

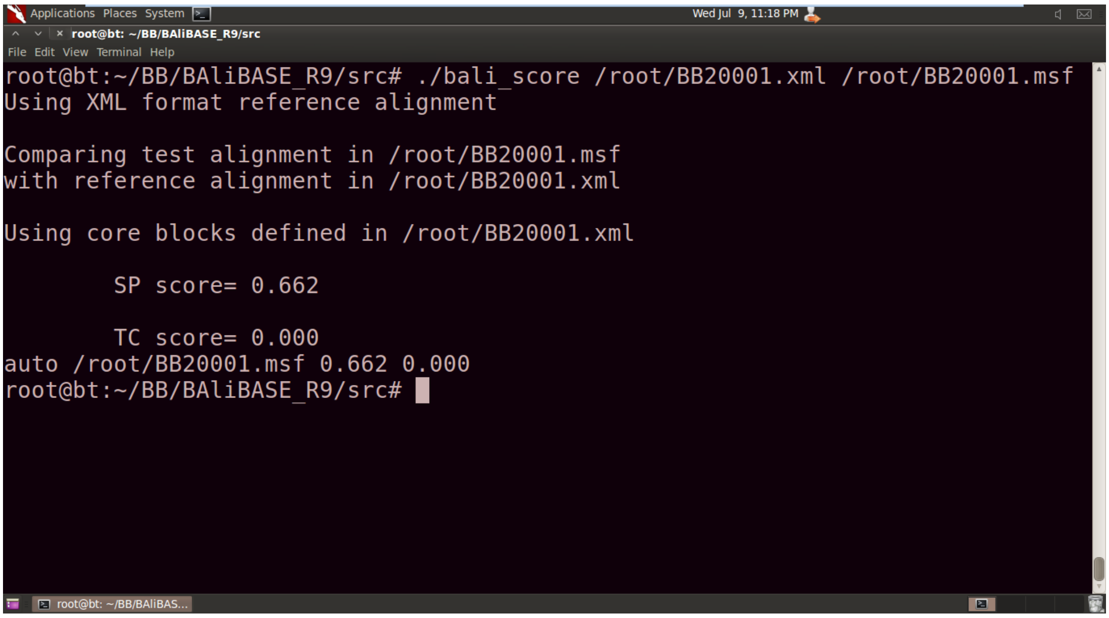

Applications Places System
Wed Jul 9, 11:12 PM
root@bt: ~/clustalw-2.1/src
File Edit View Terminal Help
root@bt: ~/clustalw-2....

```

1aab_           .......... .......... .......... .......... ..........
1e7j_A          .......... .......... .......... .......... ..........
HMGC_TETTH      .......... .......... .......... .......... ..........
2lef_A          .......... .......... .......... .......... ..........
HMGA_CHITE      .......... .......... .......... .......... ..........
HMGL_VICFA      .......... .......... .......... .......... ..........
HMGL_SOYBN      .......... .......... .......... .......... ..........
HMGH_STRPU      .......... .......... .......... .......... ..........
SP10_GORGO      .......... .......... .......... .......... ..........
HMGL_TRYBR      .......... .......... .......... .......... ..........
HG12_CAEEL      .......... .......... .......... .......... ..........
HMG2_DROME      .......... .......... .......... .......... ..........
SSRP_ARATH      .......... ....MADGHS FNNISLSGRG GKNPGLLKIN SGGIQWKKQG
SSTP_CATRO      .......... ....MADGHL FNNITLGGRG GTNPGQLRVH SGGILWKKQG
SSRP_CHICK      .......... .......... .......... .......... ..........
SSRP_CAEEL      .......... .....MAELQ FPGVYVEDVG HLAFGTLKLT EKSLNFKGDK


Press [RETURN] to continue or  X  to stop:
```

Figure 4.11: Gaps are shown by ".".

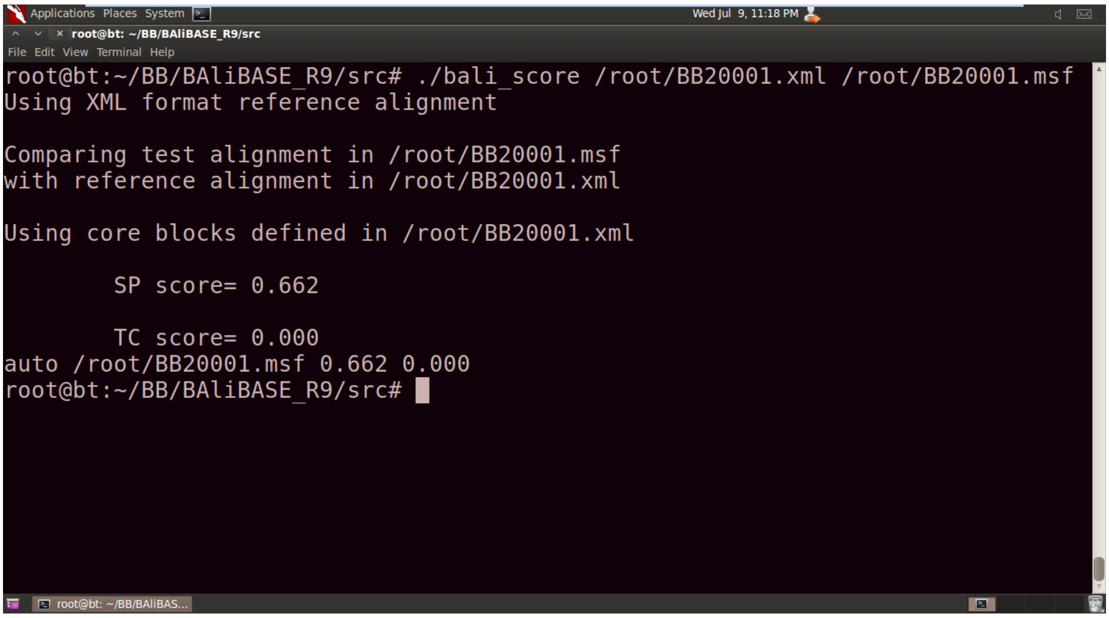
4.3. Analysis using BAliBASE

BAliBASE is a Linux tool, and it required XML parser for running in Linux environment. Configure command checks the compatibility of BAliBASE with system. After the installation of XML parser user can install BAliBASE by using make install command on terminal. Syntax of BAliBASE is "*bali_score [reference alignment] [test alignment] [-v]*" and [-v] is used for verbose mode.


Applications Places System
Wed Jul 9, 11:18 PM
root@bt: ~/BB/BAliBASE_R9/src
File Edit View Terminal Help
root@bt: ~/BB/BAliBAS...

```
root@bt:~/BB/BAliBASE_R9/src# ./bali_score /root/BB20001.xml /root/BB20001.msf
Using XML format reference alignment

Comparing test alignment in /root/BB20001.msf
with reference alignment in /root/BB20001.xml

Using core blocks defined in /root/BB20001.xml

        SP score= 0.662

        TC score= 0.000
auto /root/BB20001.msf 0.662 0.000
root@bt:~/BB/BAliBASE_R9/src# 
```

Figure 4.12: Output of original ClustalW checked by BAliBASE.


Applications Places System
Wed Jul 9, 11:18 PM
root@bt: ~/BB/BAliBASE_R9/src
File Edit View Terminal Help
root@bt: ~/BB/BAliBAS...

```
root@bt:~/BB/BAliBASE_R9/src# ./bali_score /root/BB20001.xml /root/BB20001.msf
Using XML format reference alignment

Comparing test alignment in /root/BB20001.msf
with reference alignment in /root/BB20001.xml

Using core blocks defined in /root/BB20001.xml

        SP score= 0.662

        TC score= 0.000
auto /root/BB20001.msf 0.662 0.000
root@bt:~/BB/BAliBASE_R9/src# 
```

Figure 4.13: Output of modified ClustalW checked by BAliBASE.

BAliBASE provides program for checking accuracy of output of alignment using bali_score function. There are SP and TC score shown in Figure 4.12 and 3.13 for original ClustalW tool and modified tool respectively. BAliBASE analyzes the alignment on the basis of two scores. First is SP-score which is also called sum of pairs score and second is TC score which is also called total columns score.

CHAPTER 5
EXPERIMENTAL RESULTS AND DISCUSSION

This chapter describes the results and comparison of results with the previous tool. Results are compared by BAliBASE, which is a benchmark for multiple sequence alignment. Complexity of modified tool is given in this chapter.

5.1. Experimental Results

SCS is applied in the original ClustalW2 tool of Progressive alignment. Distance matrix that previously generated by alignment score is now generated by SCS. In original ClustalW2.1 tool, a guide tree is generated by the distance matrix which is generated through alignment score. In modified ClustalW2.1, a distance matrix is generated by the length of SCS of two strings. The length of SCS is divided by the sum of the length of two corresponding strings. The Full pairwise alignment code is replaced by SCS code. The modified ClustalW2.1 tool is checked on the basis of BAliBASE 3.0 for accuracy.

The sequences of RV11 and RV 12 are used for comparing modified ClustalW2 and original ClustalW2.BAliBASE 3.0 is used for comparing reference alignment file in XML format with test alignment in MSF format. The comparison is based on Sum of Pairs (SP) scores and the Total Column (TC) scores.

RV11contains 38 datasets for benchmarking of accuracy of methods. All datasets contain sequence of protein. Datasets are made of more than 6 highly divergent sequences. It contains less than 20 percent pair-wise identity. All sequences are from distant identity.

There are 38 results BB110001-38 for 38 datasets of RV11 belongs to BAliBASE 3.0. Original ClustalW2.1 and ClustalW2.1 modified using shortest common supersequence are compared on the basis of accuracy. Accuracy based on SP scores and TC score is given in table for both original and modified tools. Readers can check enhancement by comparing 3rd columns with 5th columns and 4th and 6th columns. 3rd and 5th columns represent the SP scores of original and modified tool respectively. 4th and 6th represent the TC scores of original and modified tool respectively (Table 5.1).

Table 5.1: SP and TC Score Comparison of RV11 datasets.

Sr. No.	**Dataset**	**Original ClustalW2.1**		**Modified ClustalW2.1**	
		SP Score	***TC Score***	***SP Score***	***TC Score***
1	BB11001	0.942	0.91	1	1
2	BB11002	0.3	0	0.546	0
3	BB11003	0.402	0.2	0.699	0.53
4	BB11004	0.236	0	0.188	0
5	BB11005	0.376	0	0.434	0.14
6	BB11006	0.188	0	0.307	0
7	BB11007	0.633	0.46	0.666	0.35
8	BB11008	0.701	0.59	0.643	0.61
9	BB11009	0.342	0	0.339	0
10	BB11010	0.363	0	0.363	0
11	BB11011	0.239	0	0.357	0.19
12	BB11012	0.916	0.85	0.931	0.89
13	BB11013	0.1	0	0.1	0
14	BB11014	0.765	0.55	0.818	0.68
15	BB11015	0.743	0.62	0.685	0.47
16	BB11016	0.394	0	0.515	0
17	BB11017	0.763	0.64	0.754	0.6
18	BB11018	0.511	0	0.635	0.33
19	BB11019	0.6	0.18	0.587	0.15
20	BB11020	0.612	0.3	0.718	0.45
21	BB11021	0.252	0	0.252	0
22	BB11022	0.025	0	0.273	0
23	BB11023	0.444	0.13	0.556	0.31
24	BB11024	0.201	0	0.178	0
25	BB11025	0.114	0	0.114	0
26	BB11026	0.311	0	0.33	0
27	BB11027	0.365	0	0.341	0

Sr. No.	**Dataset**	**Original ClustalW2.1**		**Modified ClustalW2.1**	
		SP Score	***TC Score***	***SP Score***	***TC Score***
28	BB11028	0.578	0	0.649	0
29	BB11029	0.444	0.35	0.503	0.47
30	BB11030	0.389	0.05	0.43	0.18
31	BB11031	0.402	0	0.335	0
32	BB11032	0.721	0.5	0.703	0.35
33	BB11033	0.29	0	0.494	0
34	BB11034	0.394	0	0.389	0
35	BB11035	0.649	0.51	0.619	0.48
36	BB11036	0.542	0.25	0.586	0.3
37	BB11037	0.435	0.17	0.432	0.16
38	BB11038	0.557	0	0.554	0

RV11contains 44 datasets for benchmarking of accuracy of methods. All datasets contain orphan sequence. Datasets are made many similar sequences. It contains more than 40 percent similarity.

In Table 5.2, there are 44 results BB120001-44 for 44 datasets of RV12 belongs to BAliBASE 3.0. Accuracy based on SP scores and TC scores is given in table for both original and modified tools. Readers can check enhancement by comparing 3rd columns with 5th columns and 4th and 6th columns. 3rd and 5th columns represent the SP scores of original and modified tool respectively. 4th and 6th represent the TC scores of original and modified tool respectively.

Table 5.2: SP and TC Score Comparison of RV12 datasets.

Sr. No.	**Dataset**	**Original ClustalW2.1**		**Modified ClustalW2.1**	
		SP Score	***TC Score***	***SP Score***	***TC Score***
1	BB12001	0.845	0.65	0.829	0.66
2	BB12002	0.863	0.74	0.898	0.74
3	BB12003	0.865	0.59	0.932	0.89
4	BB12004	0.855	0.55	0.967	0.82
5	BB12005	0.922	0.7	0.914	0.74

Sr. No.	Dataset	Original ClustalW2.1		Modified ClustalW2.1	
		SP Score	***TC Score***	***SP Score***	***TC Score***
6	BB12006	0.959	0.92	0.959	0.92
7	BB12007	0.884	0.7	0.861	0.7
8	BB12008	0.932	0.81	0.91	0.78
9	BB12009	0.919	0.79	0.891	0.751
10	BB12010	0.913	0.81	0.93	0.83
11	BB12011	0.792	0.57	0.777	0.58
12	BB12012	0.708	0.51	0.722	0.53
13	BB12013	0.918	0.81	0.96	0.89
14	BB12014	1	1	1	1
15	BB12015	0.55	0.2	0.52	0.2
16	BB12016	0.841	0.68	0.892	0.76
17	BB12017	0.954	0.86	0.959	0.88
18	BB12018	0.958	0.92	0.957	0.92
19	BB12019	0.918	0.85	0.927	0.87
20	BB12020	0.957	0.91	0.957	0.91
21	BB12021	0.991	0.97	0.896	0.76
22	BB12022	0.849	0.74	0.924	0.86
23	BB12023	0.872	0.78	0.823	0.69
24	BB12024	0.943	0.88	0.966	0.94
25	BB12025	0.344	0	0.451	0
26	BB12026	0.909	0.72	0.895	0.66
27	BB12027	0.866	0.58	0.937	0.77
28	BB12028	0.855	0.62	0.904	0.76
29	BB12029	0.975	0.87	0.993	0.98
30	BB12030	0.969	0.93	0.979	0.95
31	BB12031	0.833	0.72	0.86	0.74
32	BB12032	0.858	0.74	0.992	0.96
33	BB12033	0.739	0.55	0.731	0.52

Sr. No.	Dataset	Original ClustalW2.1		Modified ClustalW2.1	
		SP Score	***TC Score***	***SP Score***	***TC Score***
34	BB12034	0.927	0.87	0.922	0.85
35	BB12035	0.933	0.75	0.982	0.94
36	BB12036	0.914	0.83	0.973	0.95
37	BB12037	0.889	0.7	0.869	0.68
38	BB12038	0.925	0.83	0.894	0.71
39	BB12039	0.77	0.28	0.881	0.71
40	BB12040	1	1	0.969	0.94
41	BB12041	0.742	0.58	0.786	0.6
42	BB12042	0.629	0.47	0.629	0.47
43	BB12043	0.9	0.7	0.937	0.81
44	BB12044	0.87	0.64	0.917	0.82

5.2. Complexity Analysis

Complexity of Progressive alignment using SCS is given in three steps. The first step is generating an alignment score for every pair of two nodes. There are N number of sequences, and N*(N-1)/2 is the number of the unique pairs of two sequences. SCS score is generated by an SCS length algorithm inserted in FullPairwiseAlign class. The complexity for generating alignment score for two sequences is O (L_1*L_2) where L_1 is the length of the first sequence and L_2 is the length of the second sequence. Complexity for the first step is O (L^2N^2) where L is the length of the longest sequence and N is the total number of sequences. The second step is generating guide tree from the SCS scores. Distance between two sequences is generated through the length of SCS divided by sum of the length of both sequences. Guide tree is generated by distance matrix formed by the above procedure. Complexity for generating one node of the distance matrix is O (N^2). The complexity for generating the guide tree is O (N^3). After generating the guide tree, Progressive alignment is started. Complexity of one iteration of Progressive alignment is O ($NL+L^2$). The maximum number of iterations required for completing the alignment is N-1 as internal nodes of the guide tree. Hence the complexity of the third step of Progressive alignment is O (N^2L+NL^2). The total complexity is O ($N^2L^2+N^3$).

5.3. Discussion

In Figure 5.1, a comparison between original ClustalW2.1 and modified ClustalW2.1 is represented on the basis of SP scores related to datasets of RV11. Average SP scores of RV11 for original ClustalW2.1 and modified ClustalW2.1 is 0.45366 and 0.50061. There is improvement of 10%.

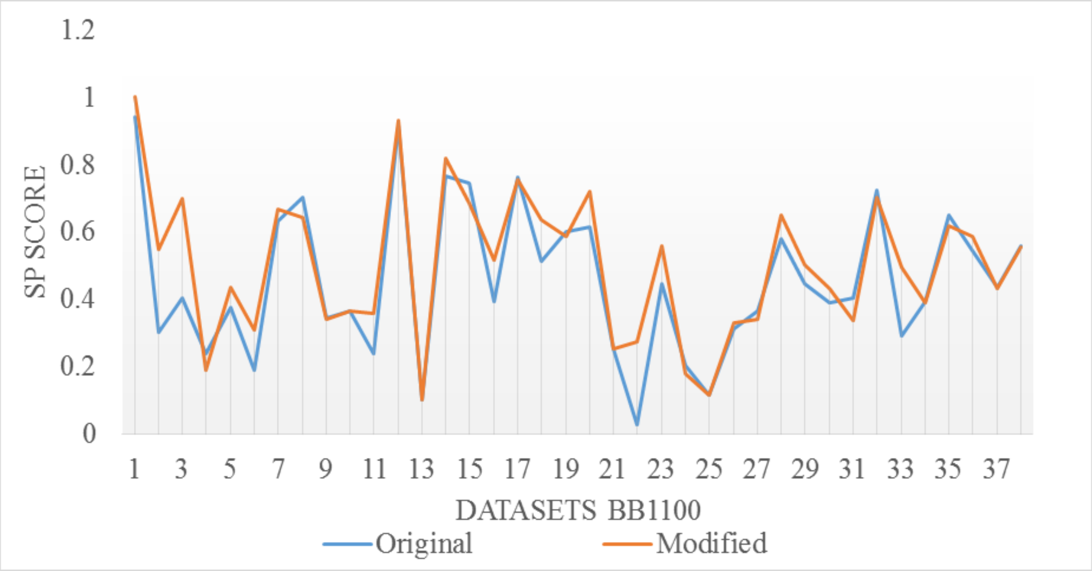


Figure 5.1: Line chart showing SP scores comparison between original and modified ClustalW tool for Datasets of RV11.

Figure 5.2 represent the TC score comparison for RV11. Average TC scores of RV11 for original ClustalW2.1 and modified ClustalW2.1 is 0.19105 and 0.22737. There is improvement of 19%. Improvement is very high for divergent type of sequences according to TC score.

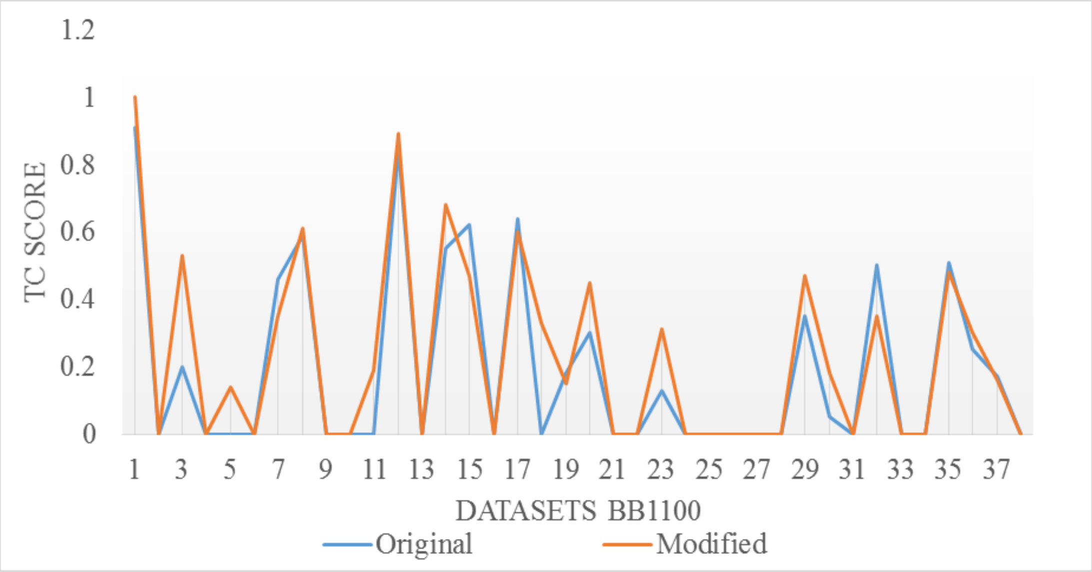


Figure 5.2: Line chart showing TC scores comparison between original and modified ClustalW tool for Datasets of RV11.

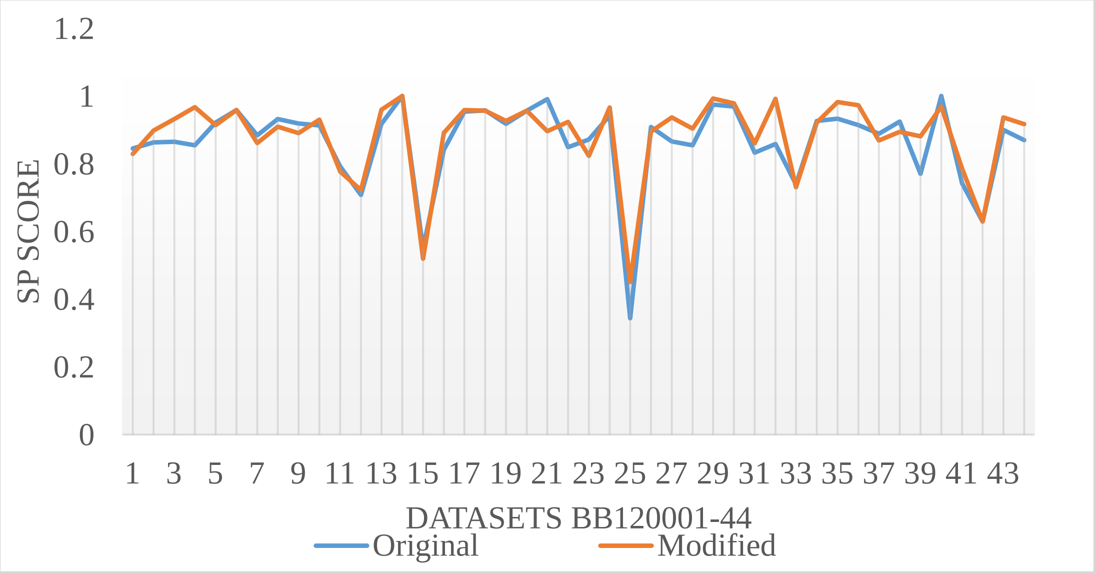


Figure 5.3: Line chart showing SP scores comparison between original and modified ClustalW tool for Datasets of RV12.

In Figure 5.3, a comparison between original ClustalW2.1 and modified ClustalW2.1 is represented on the basis of SP scores related to datasets of RV12. Average SP scores of RV12 for original ClustalW2.1 and modified ClustalW2.1 is 0.86489 and 0.88346. There is improvement of 2%.

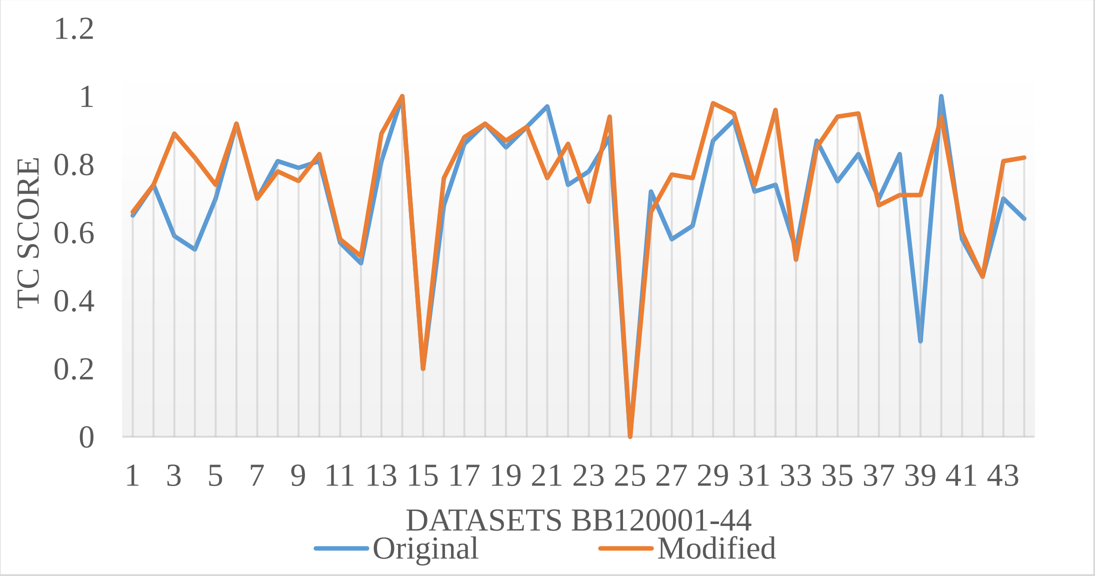


Figure 5.4: Line chart showing TC score comparison between original and modified ClustalW tool for Datasets of RV12.

In Figure 5.4, a comparison between original ClustalW2.1 and modified ClustalW2.1 is represented on the basis of TC scores related to datasets of RV12. Average SP scores of RV12 for original ClustalW2.1 and modified ClustalW2.1 is 0.71182 and 0.76002. There is improvement of 7%. Hence the accuracy for RV12 according to TC score is improved by modified ClustalW2.1.

Modified ClustalW2.1 provide better accuracy for datasets of RV11. RV11 contains highly divergent sequences. Modified ClustalW2.1 also provide better accuracy for datasets of RV12. RV12 contains highly similar sequences. This proved that the modified tool provides a better solution for both type sequences.

The complexity of the modified tool is nearly same as the original tool because complexity for finding SCS is O (N^2) for one pair of sequences, and there are N*(N-1)/2 pairs. The complexity for generating distance matrix is O (N^2L^2). Modified tools have the same complexity as existing ClustalW tool. The complexity of modified ClustalW is O ($N^2L^2+N^3$).

Above discussion proved that modified ClustalW 2.1 provides better accuracy for both divergent and similar type sequences with same time complexity as modified ClustalW2.1.

CHAPTER 6
CONCLUSION AND FUTURE SCORE

6.1 Conclusion

Progressive alignment is alignment technique using pairwise alignment and guide tree guides the alignment for generating complete solution. Guide tree in tradition progressive alignment is generated by alignment scores between sequences. Guide tree generated by alignment score does not provide an optimal solution because guide tree generated by alignment score have problem of local maximum. Guide tree can also be generated by other methods like LCS and in some cases it gives a better result. Accuracy of alignment also depend on the length of sequences because the guide tree is generated by distance scores between sequences and distance score is calculated using alignment score divided by length of the shorter sequence.

6.2 Thesis Contribution

The guide tree formed by SCS gives better results in Progressive alignment. SCS provides information about similarity of sequences without considering the gap penalty. Alignment score does not give better results as the gaps are also considered in normal Progressive alignment. But in Progressive alignment using SCS does not consider gaps in the generation of the guide tree hence gives better results according to similarity. SCS guide tree is generated using distance score of SCS and distance score of SCS is calculated by length of SCS divided by sum of the lengths of both sequences. Progressive alignment using SCS considers lengths of both sequences in pair of two sequences and gives better results. There is a remarkable improvement in TC and SP scores. The complexity of the modified tool is O ($N^2L^2+N^3$). Alignment results generated by modified tool are checked by BAliBASE benchmark and compared with original ClustalW tool.

6.3 Future Work

There is scope of improvement for special cases. Differences in lengths can measure by the standard deviation, and different variants of SCS can be used according to the standard deviation. If the standard deviation is large, then the length of SCS is divided by the sum of the lengths of two corresponding strings otherwise the length of SCS is divided by the length of bigger string. Distance scores calculated by standard deviation may provide better guide tree and this guide tree guides the alignment towards better alignment solution.

REFERENCES

[1] J. Thompson, G. Desmond, and J. Toby. “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice,” Nucleic acids research, vol. 22 no. 22, pp. 4673-4680, 1994.

[2] F. Corpet, "Multiple sequence alignment with hierarchical clustering." Nucleic acids research, vol. 16 no. 22, pp. 10881-10890, 1988

[3] “Needleman-Wunsch algorithm,” [online] Available: http://www.google.com, [Accessed: 7 march, 2014].

[4] J. Thompson, T. Gibson, F. Plewniak, F. Jeanmougin, and D. Higgins, “The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools,” Nucleic acids research, vol. 25 no. 24, pp. 4876-4882, 1997.

[5] R. Chintapalli, A. Kumar, and L. Parayitam, “Multiple sequence alignment a quick tour,” 15th International Conference on Advanced Computing Technologies (ICACT), pp. 1-6, 2013.

[6] S. Kumar, K. Tamura, and M. Nei., “MEGA3: integrated software for molecular evolutionary genetics analysis and sequence alignment,” Briefings in bioinformatics, vol. 5 no. 2, pp. 150-163, 2004.

[7] B. Modrek, A. Resch, C. Grasso, and C. Lee, "Genome-wide detection of alternative splicing in expressed sequences of human genes,” Nucleic acids research, vol. 5 no. 2, pp. 2850-2859, 2001.

[8] M. Muramatsu, K. Kazuo, F. Sidonia, Y. Shuichi, S. Yoichi, and H. Tasuku, “Class switch recombination and hypermutation require activation-induced cytidinedeaminase (AID), a potential RNA editing enzyme,” Cell 102, no. 5, pp. 553-563, 2000.

[9] D. Karolchik, R. Baertsch, M. Diekhans, T. Furey, A. Hinrichs, T. Lu, and J. Kent, “The UCSC genome browser database,” Nucleic acids research, vol. 31 no. 1, pp. 51-54, 2003.

[10] S. Hosni, A. Mokaddem, and M. Elloumi, “A new progressive multiple sequence alignment algorithm,” 23rd International Workshop on Database and Expert Systems Applications (DEXA), pp. 195-198, 2012.

[11] S. Isaza, F. Sanchez, G. Gaydadjiev, A. Ramirez, and M. Valero, “Scalability analysis of progressive alignment on a multicore,” International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 889-894, 2010.

[12] E. Olson, J. Leonard, and S. Teller, “Fast iterative alignment of pose graphs with poor initial estimates,” IEEE International Conference on Robotics and Automation, pp. 2262-2269, 2006.

[13] M. Orobitg, J. Lladós, F. Guirado, F. Cores, and C. Notredame, “Scalability and accuracy improvements of consistency-based multiple sequence alignment tools,” In Proceedings of the 20th European MPI Users' Group Meeting (EuroMPI '13). ACM, pp. 259-264, 2013.

[14] K. Katoh, K. Misawa, K. Kuma, and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform," Nucleic acids research vol. 30, pp. 3059-3066, 2002.

[15] A. Fabrice, et al. "Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee," Nucleic acids research vol. 34, pp. 604-608, 2006.

[16] K. Arnold, L. Bordoli, J. Kopp, and T. Schwede, “The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling,” Bioinformatics vol. 22 no. 2, pp. 195-201, 2006.

[17] F. Naznin, R. Sarker, and D. Essam,“Progressive alignment method using genetic algorithm for multiple sequence alignment,” Evolutionary Computation, IEEE Transactions on 16.5 , pp. 615-631, 2012

[18] K. Simons, C. Kooperberg, E. Huang, and D. Baker, “Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions,” Journal of molecular biology, vol. 268 no. 1, pp. 209-225, 1997.

[19] A. Löytynoja, and N. Goldman, “Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis,” Science vol. 320 no. 5883, pp. 1632-1635, 2008

[20] J. Besemer, A. Lomsadze, and M. Borodovsky, “GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions,” Nucleic Acids Research, vol. 29 no. 12, pp. 2607-2618, 2001.

[21] J. Thompson, P. Koehl, R. Ripp, and O. Poch, "BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark," Proteins: Structure, Function, and Bioinformatics 61(1), pp. 127-136, 2005.

[22] M. Middendorf, "More on the complexity of common superstring and supersequence problems," Theoretical Computer Science 125(2), pp. 205-228, 1994.

[23] S. Lalwani, R. Kumar, and N. Gupta. "A study on inertia weight schemes with modified particle swarm optimization algorithm for multiple sequence alignment," Sixth International Conference on Contemporary Computing (IC3) IEEE, pp. 283-288, 2013.

[24] R. Edgar, "MUSCLE: a multiple sequence alignment method with reduced time and space complexity," BMC bioinformatics 5(1), pp. 113-131, 2004.

[25] Srabanti Maji and Deepak Garg, "Hybrid Approach using SVM and MM2 in Splice Site Junction Identification," Current Bioinformatics Bentham Science 9(1), pp. 76-85, 2014.

[26] S. Needleman, and C. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” Journal of molecular biology, vol. 48 no. 3, pp. 443-453, 1970

[27] Srabanti Maji and Deepak Garg, “Hidden Markov Model for Splicing Junction Sites Identification in DNA Sequences,” Current Bioinformatics Bentham Science, 8(3), pp. 369-379, 2013.

[28] Srabanti Maji, Deepak Garg,” Progress in gene prediction: Principles and challenges,” Current Bioinformatics Bentham Sciences, 8(2), 2013.

[29] F. Smith, and M. Waterman, “Identification of common molecular subsequences,” Journal of molecular biology vol. 147 no. 1, pp. 195-197, 1981.

[30] D. Feng, and R. Doolittle, “Progressive sequence alignment as a prerequisitetto correct phylogenetic trees,” Journal of molecular evolution, vol. 25 no. 4, pp. 351-360, 1987.

[31] G. Higgins, and P. Sharp, “CLUSTAL: a package for performing multiple sequence alignment on a microcomputer,” Gene vol. 73 no. 1, pp. 237-244, 1988.

[32] J. Thompson, F. Plewniak, and O. Poch. “BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs,” Bioinformatics, vol. 15 no. 1, pp. 87-88, 1999.

[33] C. Notredame, D. Higgins, and J. Heringa, “T-Coffee: A novel method for fast and accurate multiple sequence alignment,” Journal of molecular biology, vol. 302 no. 1, pp. 205-217, 2000.

[34] Cooper, Gregory M., Senthil AG Singaravelu, and ArendSidow. "ABC: software for interactive browsing of genomic multiple sequence alignment data." BMC bioinformatics vol. 5 no.1, pp. 192-196, 2004

[35] Nasser, Sara, Gregory L. Vert, Monica Nicolescu, and Alison Murray, “Multiple sequence alignment using fuzzy logic,” IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2007, CIBCB'07 IET, 2007.

[36] P. Church, A. Goscinski, K. Holt, M. Inouye, A. Ghoting, K. Makarychev, and M. Reumann, “Design of multiple sequence alignment algorithms on parallel, distributed memory supercomputers,” 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, IEEE, 2011.

[37] L. Anderson, L. Catherine, C. Strope, and E. Moriyama, “SuiteMSA: visual tools for multiple sequence alignment comparison and molecular sequence simulation,” BMC bioinformatics vol. 12 no. 1, pp. 184-196, 2011.

[38] D. Macedo, A. Emerson, A. Magalhaes, G. Pfitscher, and A. Boukerche, “Hybrid MPI/OpenMP strategy for biological multiple sequence alignment with DIALIGN-TX in heterogeneous multicore clusters,” 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), IEEE, 2011.

[39] Naznin, Farhana, RuhulSarker, and Daryl Essam. "Progressive alignment method using genetic algorithm for multiple sequence alignment." IEEE Transactions on Evolutionary Computation, vol. 16 no. 5, pp. 615-631, 2012.

[40] F. Ortuno, J. Florido, J. Urquiza, H. Pomares, A. Prieto, and I. Rojas, “Optimization of multiple sequence alignment methodologies using a

multiobjective evolutionary algorithm based on NSGA-II," 2012 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2012.

[41] K. Nguyen, and T. Ropinski, "Large-scale multiple sequence alignment visualization through gradient vector flow analysis." IEEE Symposium on Biological Data Visualization (BioVis), IEEE, 2013.

[42] A. Mokaddeml, and M. Elloumi, "Motalign: A Multiple Sequence Alignment Algorithm Based on a New Distance and a New Score Function," 24th International Workshop on. IEEE Database and Expert Systems Applications (DEXA), 2013.

[43] T. Jiang, and M. Li, "On the approximation of shortest common supersequences and longest common subsequences," SIAM Journal on Computing, vol. 24 no. 5, pp. 1122-1139, 1995

LIST OF PUBLICATION

[1] Ankush Garg and Deepak Garg, "Progressive alignment using shortest common supersequence," *3rd international conference on Advances in computing, communication and informatics (ICACCI'14) IEEE*, 24 – 27 September 2014, Greater Noida, India, 2014 (**Accepted**).