

SEQUENCE ALIGNMENT EDITOR - A NEW TOOL TO ASSIST WITH
INFERRING PHYLOGENETIC AND FUNCTIONAL
RELATIONSHIPS IN BIOLOGICAL DATA

by

RAMYA RAGHUKUMAR

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2007

Copyright © by Ramya Raghukumar 2007

All Rights Reserved

ACKNOWLEDGEMENTS

I thank my supervising professor Dr.Nikola Stojanovic, for constantly encouraging and motivating me and also for his invaluable knowledge that he has passed on me to during the course of my Master's studies. I also thank Mr. David Levine and Dr. Manfred Huber for taking interest in my research and consenting to participate in my defense committee. I would like to thank Dr.Bahram Khalili and Mr. Mike O' Dell for their academic advising during my Master's program. I thank the CSE department for funding me throughout my thesis work.

Finally I would like to express deep gratitude to my parents for encouraging and inspiring me and for supporting my graduate studies. I am also grateful to my sisters and grand parents for their support and guidance. I also take thank my friends who have stood by me throughout my endeavors.

Last but not the least, I thank GOD ALMIGHTY, who has given me strength and walked me through every stage of my life.

I dedicate this work to my family and GOD.

July 19, 2007

ABSTRACT

SEQUENCE ALIGNMENT EDITOR-A NEW TOOL TO ASSIST WITH INFERRING PHYLOGENETIC AND FUNCTIONAL RELATIONSHIPS IN BIOLOGICAL DATA

Publication No. _____

Ramya Raghukumar, M.S.

The University of Texas at Arlington, 2007

Supervising Professor: Dr.Nikola Stojanovic

Biologists study evolution, discover functional and structural information in genomic or protein data through the extensive use of sequence alignments. It is very difficult to manually align long regions, so development of methods for this task continues to be an active area of research.

Alignment construction algorithms, based on dynamic programming, approach the problem from a mathematical perspective. The optimal alignment is computed relative to a scoring scheme. The resulting layout is guaranteed to be mathematically optimal, though not necessarily biologically meaningful. Sequence alignments usually account only for single letter substitutions and relatively short indels, representing the

latter as gaps. Other possible evolutionary scenarios like rearrangements and inversions are generally not considered, although that situation started changing recently.

In a multiple sequence alignment, the most popular method for reigning in complexity is by using a progressive approach. Progressive alignment techniques can incorporate limited evolutionary information in the form of a phylogenetic tree while building alignments. Except for that, they generally do not make use of knowledge that shed light on the sequences in a biological context, although there are some notable exceptions. Progressive techniques may be implemented in iterative refining steps; however they still remain both computationally and biologically approximate.

At present, biologists are often forced to manually adjust the alignments built through automated means. These adjustments include the placement of sites of experimentally confirmed homology or characteristic structural features in conformation, when their correspondence is not well reflected in the sequences, thus misguiding the automated (mathematical) optimization process.

Many tools for alignment visualization provide extensive annotation facilities, but in most cases they are passive. Some feature editors are available; however these are prominently sequence-only editors. Since they would permit sequences to be removed, new bases introduced and/or existing ones deleted, the entire concept is somewhat at odds with the idea of alignment editing, in which the sequences and order of residues in individual sequences should not be disturbed.

With these issues in mind, we have undertaken the design of an editor which would facilitate post-processing of sequence alignments. The core editor features

provide for drag-and-drop movement of regions within the aligned sequences, followed by the realignment of the affected area or a broader context, depending on the user's selection. The realignment is done through the use of external freely available software. However, it should be noted that at any particular installation not all choices will be supported, as the external software packages may not run on every platform. The region movement and realigning can be repeated as many times as necessary.

This utility has been developed using Java Swing library. It can be run on any installation which supports Java, executed locally on the user's machine.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
ABSTRACT	iv
LIST OF ILLUSTRATIONS.....	xi
Chapter	
1. INTRODUCTION.....	1
1.1 Need for sequence alignments	7
1.2 Sequence alignment techniques	8
1.2.1 Dot Matrix.....	9
1.2.2 Dynamic Programming	10
1.2.2.1 Global pairwise alignment.....	11
1.2.2.2 Local pairwise alignment.....	14
1.2.2.3 Dynamic programming with affine gap penalties.....	15
1.2.2.4 Drawbacks of dynamic programming.....	17
1.2.3 Heuristic techniques for alignment construction.....	17
1.2.3.1 Heuristic local pairwise alignment	18
1.2.3.2 Progressive alignment technique for global multiple sequence alignment.....	20

2. PREVIOUS WORK.....	24
2.1 Alignment editors and visualization tools	24
2.2 Available tools.....	28
2.2.1 VISTA	28
2.2.2 CINEMA.....	30
2.2.3 SeaView.....	32
2.2.4 Jalview.....	33
3. SEQUENCE ALIGNMENT EDITOR.....	36
3.1 Summary.....	36
3.2 Implementation platform	36
3.3 Editor layout	37
3.4 Display of annotated regions	38
3.5 Alignment editing	39
4. EXAMPLES.....	45
4.1 Input.....	45
4.2 Hox genes.....	45
4.3 Alignment before editing.....	45
4.4 Alignment after post processing.....	48
5. FUTURE WORK.....	49
Appendix	
A. INSTRUCTION MANUAL	51

B. USING SEQUENCE ALIGNMENT EDITOR	71
REFERENCES	73
BIOGRAPHICAL INFORMATION.....	77

LIST OF ILLUSTRATIONS

Figure	Page
1.1 Central dogma of molecular biology	3
1.2 Example of a DNA sequence taken from GenBank	3
1.3 Example of a protein sequence taken from GenBank	4
1.4 Illustration of an application of Dot Matrix technique	10
1.5 Constructing cell pointers for Needleman-Wunsch	13
1.6 Trace back for Needleman-Wunsch	14
1.7 Constructing cell pointers for Smith-Waterman	15
1.8 Trace back for Smith-Waterman	15
1.9 A Phylogenetic tree in progressive alignment	22
2.1 A screen shot of the Vista Browser	30
2.2 A screen shot of the CINEMA viewer	32
2.3 A screen shot of the SeaView editor.....	33
2.4 A screen shot of the Jalview editor	35
3.1 Screen shot of editor layout	38
3.2 Annotation legend	39
3.3 Selection of sites in sequences 1 and 2	40
3.4 Placement of sites in one column	40
3.5 Gaps within the area of conformation	41

3.6 Screen shot showing the area of conformation and disturbed regions	43
4.1 Screen shot of the Hox A input alignment, at the beginning of exon1 of HoxA11 gene, in the human sequence	46
4.2 Screen shot showing selection of “ATG” triplet in mouse sequence	47
4.3 Screen shot after placing “ATG” triplet of mouse in conformation with the “ATG” triplet of other sequences.....	47
4.4 Hox A Alignment after post processing	48
A1 File formats supported by Sequence Alignment Editor	54
A2 File menu.....	56
A3 Search menu	57
A4 Find option	58
A5 Go To option	60
A6 View menu	61
A7 Fix Region menu	63
A8 Selection menu	63
A9 lock option in the pop up menu.....	65
A10 mark till option.....	66
A11 Movement menu.....	67
A12 pop up menu	68
A13 Font menu.....	69
A14 Realignment menu	70

CHAPTER 1

INTRODUCTION

The notion of biological sequences usually comprise DNA, RNA and protein sequences.

DNA (Deoxyribonucleic acid)

DNA can be viewed as the blueprint for an organism. It conveys the information needed to construct and understand its development and functioning. Chemically, DNA is a polymer made up of molecules called nucleotides. There are four different nucleotides, each made up of a combination of a phosphate group, a pentose carbon deoxyribose sugar and a nitrogenous base. The nucleotides are linked to each other by covalently joining adjacent sugar rings between the third and fifth carbon atoms, through the phosphate group. Thus the entire polymer consists of a sugar-phosphate backbone with nitrogenous bases attached out to the backbone. These bases differentiate one type of nucleotide from another. The four bases are adenine (A), cytosine (C), guanine (G) and thymine (T). DNA is double stranded and helical in structure [29]. The two strands are antiparallel, with the top strand running from the 5' end to 3' end and the bottom strand running from the 3' end to 5' end. The 5' end indicates that the fifth carbon atom of the deoxyribose sugar is free at that end of the sugar-phosphate backbone and the 3' end indicates that the third carbon atom is free at that end.

RNA (Ribonucleic acid)

RNA is similar to DNA in composition, except for the presence of ribose sugar instead of deoxyribose and presence of the uracil (U) base instead of thymine (T). Moreover, RNA is normally single stranded polymer. It is synthesized from DNA, with the help of enzymes called RNA polymerases. Most RNA thus produced is used to generate a protein.

Proteins

Proteins are the building blocks of life. They can be broadly classified into those providing structural components of a cell and those with enzymatic activity and signaling compounds. Proteins of all living organisms are made up of twenty amino acids. A protein is synthesized based on a template enclosed in its gene. Gene is a stretch of DNA that produces a protein. The gene is first transcribed to an mRNA. The mRNA is translated to a protein.

Central dogma of molecular biology

The central dogma describes the way genetic information is used to create proteins essential for the functioning of the organism. The flow of information is generally unidirectional although there are many instances which do not follow this simplistic model.

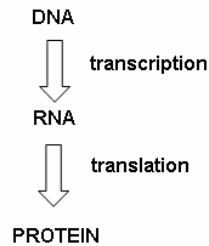


Figure 1.1 Central dogma of molecular biology

DNA sequences

A DNA sequence is represented as a chain of combination of four letters A, C, G and T that representing the nucleotide bases. Example of a DNA sequence, taken from A GenBank sample record is shown in Figure 1.2 below. It is an expressed sequence tag and corresponds to accession number CU453739.

GenBank is the National Institutes of Health genetic sequence database, an annotated collection of all publicly available DNA sequences maintained by the US National Center for Biology Information. (<http://www.ncbi.nlm.nih.gov/>)

SEQUENCE

```

CCTCATGGACTCTTCCCTTACTGTGCATCATCCTCCTCTTCATCACTGTCTCCCTTCTTGG
CATTCTTGCCATTCTTTGCCCCCTTGGCTGGGATGGCAGCACCCCTTCTTACCAGGAGTTG
CTACCAATGCTTTGCCTGGTGTGGCTCCCTTCTTGCCAGGTGTGGTAACCTGCTTTGGCTG
GTGTAACCTGTCTTCTTGGCAGGTGTTGCTGCTGCCTTTTTGCCTGGAGTGACAGCTGCTT
TCTTGGCTGGTGTGGCAACTGCAACCTTTTTTGTGGGGAACGACCACCTTCTTTGCTG
AGGTTGCAGCAGCCTTCTTGCCTTCTTCTGAGGTATGACGACCTCTTCTCCACTGCTAT
CATCTTCTTCATCTTCTGACATTTCCTCATCTTCACTATCTTCTTCTACCTCCTTTGGAG
GAGGAGCCATTTTCTTGGGGTCACCTTGATTTTACCTGCCTTCGCGAGCTTCACCATGA
TGGCGGCGGAGTATGTAGCGGCTCATGGC
  
```

Entry Created: Aug 3 2007
 Last Updated: Aug 3 2007

Figure 1.2 Example of a DNA sequence taken from GenBank, accession no. CU453739

Protein sequences

A protein sequence can be viewed as a chain of the twenty letters which represent the constituent amino acids of that protein.

Example protein sequence: Ax12p protein, GenBank ID: AAA98666.1 is shown in Figure 1.3 below. It has been taken from a sample GenBank record of species *Saccharomyces cerevisiae* (<http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html>).

```
/product="Ax12p"  
/protein_id="AAA98666.1"  
/db_xref="GI:1293615"  
/translation="MTQLQISLLLTATISLLHLVVATPYEAYPIGKQYPPVARVNESF  
TFQISNDTYKSSVDKTAQITYNCFDLPWLSFDSSSRTFSGEPSSDLLSDANTTLYFN  
VILEGTDSDSTSLNNTYQFVVTNRPSISLSSDFNLLALLKNYGYTNGKNALKLDPNE  
VFNVTDFDRSMFTNEESIVSYYGSQLYNAPLPNWLFDFSGELKFTGTAPVINSIAIPE  
TSYSFVIIATDIEGFSAVEVEFELVIGAHQLTTSIQNSLIINVTDTGNVSYDLPLNYV  
YLDDDPISSDKLGSINLLDAPDWALDNATISGSVPDELLGKNSNPANFSVSIYDTYG  
DVIYFNFEVYSTTDLFAISSLPNINATRGEWFSYYFLPSQFTDYVNTNVSLEFTNSSQ  
DHDWVKFQSSNLTLAGVFPKNFDKLSLGLKANQGSQSQELYFNIIGMDSKITHSNHSA  
NATSTRSSHSTSTSSYTSSTYTAKISSTSAATSSAPAALPAANKTSSHNKKAVAIA  
CGVAIPLGVILVALICFLIFWRRRRENPDENLPHATSGPDLNPNANKPNQENATPLN  
NPFDDDDASSYDDTSIARRLAALNTLKLDNHSATESDISSVDEKRDSLGMNTYNDQFQ  
SQSKEELLAKPPVQPPEPFFDPQNRSSSVYMDSEPAVNKSWRYTGNLSPVSDIVRDS  
YGSQKTVDTEKLFDEAPEKEKRTSRDVTMSSLDPWNSNISPSVPRKSVTPSPYNVTK  
HRNRHLQNIQDSQSGKNGITPTTMTSTSSDDFVPVKDGENFCWVHSMEDRRRPSKKRL  
VDFSNNKSNVNVGQVKDIHGRIPEML"
```

Figure 1.3 Example of a protein sequence taken from GenBank, accession no. U49845

Sequence alignments

Alignments are used to compare two or more sequences by searching for patterns of characters that occur in the same order in all sequences and the preservation of order of bases/amino acids is very important. The alignment is constructed by placing the sequences one below the other, with the corresponding characters placed in the same column. Gaps (customarily represented by the hyphen symbol) are introduced in the alignment to reflect small insertion and deletion events.

Sequence alignments can be classified based on two criteria:

1. Number of sequences.
 - a. Pairwise sequence alignments.
 - b. Multiple sequence alignments.
2. Extent of sequences aligned.
 - a. Global alignments.
 - b. Local alignments.

1. Based on number of sequences:

- a. Pairwise alignments

As the name suggests, this is an alignment of two sequences.

Example of a pair wise alignment:

Columns 1 2 3 4 5 6 7 8 9 10 11 12

Sequence 1: A C G G C T T C G A T C

 | | | | | | | | |

Sequence 2: A C G - C A A C G A T C

The vertical bars indicate alignment of identical bases. The 4th column in the alignment shows the placement of base “G” over a Gap “-“. This indicates either deletion of a base in that particular position from the second sequence or an insertion of a base in that particular position in the first sequence. The 5th and 6th columns show two different bases “T and “A” aligned over one another; these are examples of mismatched columns. Though not identical, they represent likely mutational event, from a biological perspective.

b. Multiple sequence alignments

This is an alignment of two or more sequences.

Columns : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Sequence 1: A C G C G T T C G A A A T G C G A T

Sequence 2: A C G C - T - - G A - A T G G C A -

Sequence 3: A C G C G A - - G - - A T G - G A T

- Columns 1, 2, 3, 12, 14, 17 are identical.
- Columns 5, 7, 8, 11 and 18 have gaps and indicate insertion/deletion events.
- Columns 4, 6, 16, 18 have mismatched bases, possibly a likely alignment from a biological perspective.

2. Based on the extent of sequences aligned:

a. Global alignments

Alignment is done over the entire length of the sequences including as many similar or identical regions of all the sequences in a column.

b. Local alignments

In this kind of alignment, the search for conformation stops at the boundaries of similar regions and does not extend to include more nucleotides or amino acids. This type of alignment is favoured for finding short conserved patterns of nucleotides or amino acids between

sequences. A local alignment may return one or more similar subsequences, but discards all regions that do not match well.

1.1 Need for sequence alignments

Sequence alignments are useful for discovering structural, functional and evolutionary information in genomic or protein data.

Similar sequences may have originated from a common source. A multiple alignment of a set of sequences may thus aid the reconstruction of their evolutionary history. If the sequences align very well they are likely to be derived from a common ancestor, whereas sequences poorly aligned can be at best distantly related. While investigating relationships among sequences, various scenarios should be accounted for, such as insertion and deletion mutations, rearrangements, repeats and duplication events in the genomes over a period of time.

For example, many *genes are believed to be born out of a gene duplication event. Two copies of the same ancestral gene are created and these copies accumulate mutations and may undergo several other changes to evolve different genes* [3]. Genes derived from a common ancestor are said to be homologous.

Homologous genes from the same species are called paralogs. Many genes are members of often very large families of paralogs, such as globin or hox genes in mammals. Homologous genes sharing the same function in different species are called orthologs. Some protein sequences have similar function not because of common ancestry but due to a convergence of separate evolution pathways. They are called analogs. Sequence alignments contribute significantly to studies on evolution, which are

concerned with the origin and descent of species, as well as their change, multiplication, and divergence over time. Evolutionary relationships thus inferred from sequence alignments are used in the phylogenetic or evolutionary tree prediction. [3]

A multiple sequence alignment of protein sequences may define a family of proteins. Proteins of the same family display similar biochemical functions and structural properties apart from the fact that they may have evolved from a common ancestor. When aligning an unknown protein sequence with known proteins, similar domains or active sites conserved in the alignment argue for the unknown protein sequence to be a probable member of the family of the known proteins.

For example, if the structure of a protein in the well defined multiple sequence alignment is known, it may be possible to predict which amino acids have the same spatial organization in other aligned proteins. In the case of local alignment of protein sequences, consensus information inferred may be used as probes to find sequences that are potential members of the same family. This helps proteins to be grouped structurally or functionally, providing a good base for further research. Similarly, alignment of short conserved patterns of a set of nucleotide sequences help in the detection of motifs and their functions.

1.2 Sequence alignment techniques

Sequence alignment is one of the most common tasks in bioinformatics and many techniques have been developed for their construction. In this section, we outline several major alignment techniques.

1.2.1 Dot Matrix

This is a technique for aligning and analyzing a pair of sequences [1, 3]. One sequence is written horizontally and the other sequence is written vertically. Fixed size windows of characters from both sequences are compared and if such windows are sufficiently similar they are identified by a dot symbol. A diagonal of such dots or marks identifies a broader conserved region. This method is useful for identifying long conserved patterns, inversions and repeats in sequences. Repeats are a common feature of higher eukaryotic genomes and they may be interspersed or may occur in a continuous block [2]. The window size affects the quality of the dot plot. A small window size over long sequences will introduce many dots here and there in the matrix, making the analysis difficult. On the other hand, a larger window size may eliminate the stray dots but there are chances of missing short conserved patterns as well.

Some of the drawbacks of Dot Matrix are:

1. Their analysis is mostly visual, and difficult to automate.
2. Identical residues are marked with dots. Similar matches can be scored, but good statistical measures of that can be hard to obtain.
3. Except under very unusual circumstances, it is difficult to detect the optimal alignment based on dot patterns alone.

Figure 1.4 provides an example of a dot matrix technique.

Red marks along diagonals indicate conserved regions in both sequences.

	C	G	T	T	A	G	C	C	T
A					*				
C	*						*	*	
G		*				*			
T			*	*					*
T			*	*					*
C	*						*	*	
C	*						*	*	
G		*				*			
A					*				

Figure 1.4 Illustration of an application of Dot Matrix technique

1.2.2 Dynamic Programming

In this approach, the problem of determining the final alignment is broken into sub steps. At every step an optimal alignment of growing sub-sequences is found, and this result is used in the next step, giving the final alignment in the end. Optimal alignment is determined based on a scoring scheme where the alignment with the highest score is considered optimal. A dynamic programming technique has been proven to guarantee an optimal alignment, at least mathematically.

There are two popular algorithms based on the dynamic programming approach:

Needleman-Wunsch method for Global pair wise sequence alignment.

Smith-Waterman method for Local pair wise sequence alignment.

1.2.2.1 Global pairwise alignment

Saul Needleman and Christian Wunsch developed an algorithm in 1970, to align sequences globally [22]. By far, this has been the most widely used technique for generating the optimal alignment of two sequences. Such alignments are an important source of information for biologists and they have led to significant discoveries in genomic and protein sequences.

The Needleman-Wunsch algorithm makes use of a simple scoring scheme. Each column is awarded points scored based on a match or mismatch or an introduction of a gap. Columns are scored independently, meaning at mutations are independent. For instance, one may want to reward a match with +1 points and penalize a substitution or an indel by -1.

For protein sequences, substitutions, insertions and deletion scenarios are considered differently than DNA, through the use of substitution matrices.

For example, the score for a column consisting of amino acid Alanine (“A”) in both sequences, taken from BLOSUM62 (BLOcks of Amino Acid SUBstitution Matrix) matrix is 4 [16]. Alignment column scores are additive, i.e. the sum of the scores of all columns, gives the final alignment score.

For Example,

Sequence 1 A C G C T

Sequence 2 A C - A T

Score 1 1 -1 -1 1

Final alignment score: 1

Even for small sequences, there is a large number of possible alignments. It is not practical to generate them all determine the one with the maximum score. With dynamic programming an optimal alignment is found effectively by using the optimal solutions produced at every step.

To compute the alignment using the Needleman-Wunsch method, a two dimensional matrix of size $(m+1) (n+1)$ is created, where m and n are lengths of sequences $s1$ and $s2$ respectively. Columns are numbered 0 to $m-1$ i.e. and rows are numbered 0 to $n-1$. $C(i,j)$ is the cell at the i^{th} row and j^{th} column. Every $C(i,j)$ is assigned a score $H(i,j)$, the score of the optimal alignment obtained by aligning the first i characters of sequence $s1$ with first j characters of sequence $s2$. The first row and column are filled with gap penalties because they represent gaps over characters. $A(i,j)$ denotes the alignment of first i characters of sequence $s1$ with first j characters of sequence $s2$. $S(i,j)$ represents the points awarded for a match between the i^{th} character of sequence $s1$ and j^{th} character of sequence $s2$.

Alignment $A(i,j)$ can end in one of the following 3 ways:

- i^{th} character aligned with j^{th} character.
- i^{th} character aligned with a gap
- j^{th} character aligned with a gap

Based on the above conditions, each $H(i,j)$ can be calculated as :

$$H(i,j) = \max \{$$

$$H(i-1,j-1) + S(i,j),$$

$$H(i-1,j) + \text{gap penalty},$$

$$\left. \begin{array}{l} H(i,j-1) + \text{gap penalty} \\ \end{array} \right\}$$

Even as $H(i,j)$ is calculated for each cell, a pointer to the previous step from where the current score is derived, is maintained, in order to ensure the traceback, in which the optimal alignment is constructed based on the scores.

Figure 1.5 shows the $H(i,j)$ values each cell $C(i,j)$ along with the pointers, for a hypothetical alignment of sequence “AGAC” with sequence “AGC”.

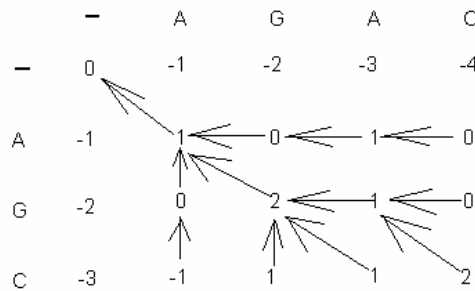


Figure 1.5 Constructing cell pointers for the Needleman-Wunsch algorithm

TraceBack

The final alignment is constructed from the scores matrix as follows:

The construction starts from the last cell i.e. $C(m,n)$. The path is traced back following the pointers from the final maximum score.

A diagonal arrow from $C(i-1,j-1)$ indicates alignment of i^{th} character with j^{th} character. A horizontal arrow from $C(i-1,j)$ indicates alignment of i^{th} character with a gap. A vertical arrow from $C(i,j-1)$ indicates alignment of j^{th} character with a gap.

Figure 1.6 shows the traceback, with the optimal alignment spelled by the path of blue lines. In this case it is:

AGAC

AG - C

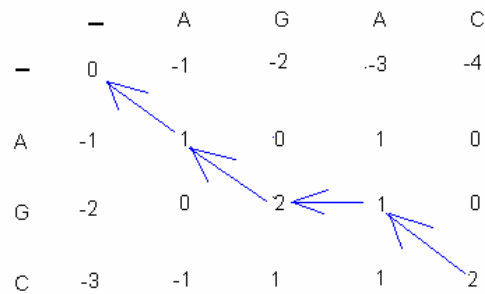


Figure 1.6 Trace back for Needleman-Wunsch

1.2.2.2 Local pairwise alignment

In 1981, Temple Smith and Michael Waterman developed an algorithm for finding local alignments, based on the Needleman-Wunsch approach, with the following modifications [28]:

The scoring matrix must not contain negative values. If a cell has a negative score, it is substituted with 0. The traceback starts from the cell with the highest score in the matrix. The traceback stops at a cell with score 0, giving the best scoring local alignment. The formula for calculating the score $H(i,j)$ at each cell $C(i,j)$ is as follows:

Score at each cell, $H(i,j) = \text{maximum}\{ H(i-1,j-1) + S(i,j),$

$H(i-1,j) + \text{gap penalty},$

$H(i,j-1) + \text{gap penalty},$

0}

A sample trace of the Smith-Waterman algorithm on sequences “TGAC” and “GAC”, with a match reward of +1 and mismatch/indel penalty of -1 is shown in figure 1.7, and the optimal alignment is shown in figure 1.8.

The final alignment is:

G A C
G A C

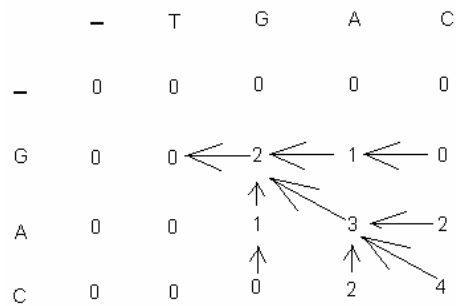


Figure 1.7 Constructing cell pointers for Smith-Waterman

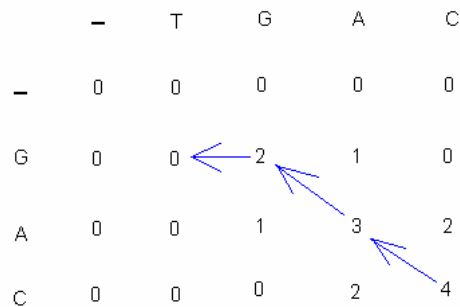


Figure 1.8 Trace back for Smith-Waterman

1.2.2.3 Dynamic programming with affine gap penalties

The methods described above used a linear gap penalty function for scoring the alignment. With linear gap penalties, each gap is considered to be an

individual mutation event. However over the evolution, it is more likely that a stretch of indels were introduced as a consequence of a single mutation event. The gaps thus need to be scored differently. When a gap is opened, it is assigned a gap opening penalty. The gaps following the newly opened gap are assigned gap extension penalty.

For example, A gap of length n , is scored as: $\text{gap_open} + (n-1)*\text{gap_extend}$

Both Needleman-Wunsch and Smith-Waterman algorithms are slightly different when using affine gap penalties. The score at each cell, $H(i,j)$ is still derived from the scores $H(i-1,j-1)$, $H(i-1,j)$ and $H(i,j-1)$ but it also depends on whether there was a gap already open in any of the sequences or not. Consequently, these different variants of $H(i,j)$ need to be maintained: $H^*(i,j)$ which corresponds to no gap open, $H^+(i,j)$ corresponding to gap open in the top sequence and $H^-(i,j)$ referencing a gap open in the bottom sequence. The formula to update the values in the Needleman-Wunsch approach is then:

$$H^*(i,j) = \max \{$$

$$H^*(i-1,j-1) + S(i, j),$$

$$H^-(i-1,j-1) + S(i, j),$$

$$H^+(i-1,j-1) + S(i, j) \}$$

$$H^-(i,j) = \max \{$$

$$H^*(i-1,j) + \text{gap_open},$$

$$\begin{aligned}
& H^-(i-1,j) + \text{gap_extension}, \\
& H^+(i-1,j) + \text{gap_open} \} \\
H^+(i,j) = \max \{ & \\
& H^*(i,j-1) + \text{gap_open}, \\
& H^-(i,j-1) + \text{gap_open}, \\
& H^+(i,j-1) + \text{gap_extension} \}
\end{aligned}$$

Some implementations of the affine scoring scheme are also prohibiting an opening of the gap in one sequence immediately after closing it in another.

The modifications to the Smith-Waterman algorithm in order to accommodate affine gap penalties are straight forward, as they require only the addition of a zero in the way analogous to our previous description.

1.2.2.4 Drawbacks of dynamic programming approaches

Both Needleman-Wunsch and Smith-Waterman algorithms are algorithm is well suited for aligning a pair of sequences of moderate length. However their performance is quadratic and genomic sequences are usually very long; so time and space constraints are a factor limiting their use in most practical situations.

1.2.3 Heuristic techniques for alignment construction

Heuristic techniques are intuitive approaches to the problem of alignment construction of long sequences and of more than two sequences. These techniques do not guarantee an optimal alignment, however they attempt to generate a reasonably good one. Since they don't provide optimal solutions, they do not take up the time and

space like dynamic programming techniques. Out of a wide variety of heuristic methods, we shall here address only the following popular ones:

- FASTA and BLAST programs for local pairwise alignment.
- Progressive alignment for global multiple sequence alignment.

1.2.3.1 Heuristic local pairwise alignment

FASTA

FASTA [9] is a technique for performing rapid pairwise alignments of protein or DNA sequences. Instead of comparing the sequences residue by residue, it searches for short patterns of length k , called k -mers. FASTA then tries to build a local alignment based on these k -mers. The query sequence is compared with all sequences in a target database and the best matched sequences and local alignments of the best matched sequences with the query are returned to the user. The length of the k -mers can be user defined, but is generally 4-6 nucleotides for DNA sequences and 1-2 amino acids for protein sequences. FASTA uses hashing method to perform the search for k -mer matches.

It creates a lookup table for each sequence and records the position of each k -mer. The relative positions of each word in the two sequences are calculated by subtracting the position of that word in the first sequence from its position in the second sequence.

BLAST

BLAST [26] technique is similar to FASTA: it increases the speed of constructing the alignments by searching for matching k -mers in the query sequence and each sequence in the target database. Like FASTA, BLAST does not search for all matching k -mers, restricting it only to significant matching patterns. By default, BLAST fixes the k value at 11 for DNA sequences and 3 for protein sequences. However the user has some leverage to change this. This length has been reported to achieve a significant hit without missing on short significant patterns. BLAST can filter out known repetitive regions in the sequences in order to avoid finding matches which are actually expected to be of much scientific interest. This is provided as an user option, though. For a protein sequence, a list of three letter words starting from position 1 till the last available position in the sequence is constructed. Each word can be matched with 8000 3-letter words taking all possible combinations of letters from the 20-letter protein alphabet. These words are evaluated for significance by scoring them using the BLOSUM 62 matrix scores. Words that score above a certain threshold value are retained. (For example, SAI (Serine-Alanine-Isoleucine) would not score well compared to itself, 12 by BLOSUM 62 [16], while CWH (Cysteine-Tryptophan-Histidine) would score high: 28 by BLOSUM 62).

This process is repeated for each 3 letter word of the query sequence. The highest scoring words are compared with sequences in the target database. Each target sequence is scanned to check for a match with the query sequence words. If a match is found, it is used as an anchor or seed for starting the construction of an ungapped local

alignment between the query and the target .The alignment is extended along both directions from the seed until the score of the local alignment falls below a threshold value. Such high scoring pair that makes the seed for alignment is called a maximal segment pair. Currently there are many different versions of BLAST, and we describe one straightforward extension below.

BLAST 2

Blast 2 is a recent version of the Blast program. It generated gapped local alignments between the query sequence and the sequence from the target database.

This method is similar to the original BLAST described above, but it looks for two significant words in a close proximity while constructing the local gapped alignment. A list of high scoring words is constructed as in the previous version, but the words are filtered based on a slightly lower threshold score. Once a pair of high scoring words is extended on either side of this composite seed, and is stopped when the alignment score falls below a certain threshold score.

1.2.3.2 Progressive alignment technique for global multiple sequence alignment

Performance of Needleman – Wunsch Algorithm

Size of matrix = $(m+1)(n+1)$

where m and n are lengths of the two sequences to be aligned.

Number of cells to be examined = $2^n - 1$

Running time = $O(mn)$

If the lengths of the sequences increase, the computational steps and running time increase exponentially. This leads to the application of progressive alignment technique for alignment construction of multiple sequences.

Progressive alignment [8] works by constructing the multiple sequence alignment in a step by step fashion. Pairwise alignments of the sequences are constructed initially, and scored. These alignment scores are used to construct a phylogenetic tree represent the evolutionary (or functional, when an evolutionary view is not appropriate) relationship among the sequences. Based on the phylogenetic tree, the sequences that are most closely related are aligned pairwise. The next closely related sequence is then aligned to the initial pairwise alignment to give a 3-way alignment. This procedure is repeated until all the sequences taken into account to give the final multiple sequence alignment.

An example of an evolutionary tree which can be used for progressively aligning sequences is given in figure 1.9. The resulting alignment can be:

A C G C T

A C - - T

A C G - T

For this alignment, mouse and rat sequences were considered first, and human has been added only at a later stage.

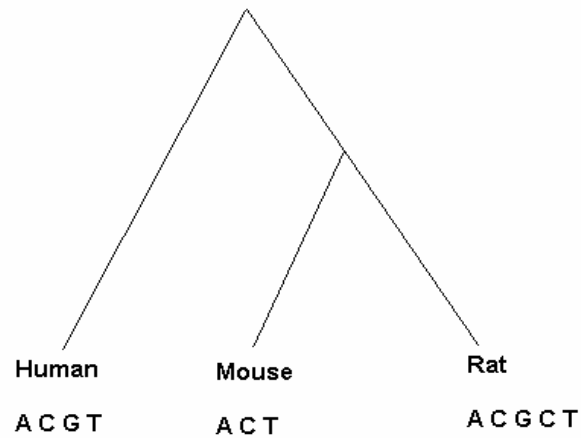


Figure 1.9 A Phylogenetic tree in progressive alignment

The quality of the final alignment depends on the initial pairwise sequence and other intermediate alignments. If the sequences are well related, chances of getting the initial pairwise sequence alignment and hence the final results accurate are high. However if the sequences are distantly related, chances of committing errors during the initial stages are high. The errors from these first stages are propagated till the end, leading to a final alignment that is inaccurate.

In addition to the problem of using a wrong initial sequence alignment, the usage of inappropriate scoring matrices and gap penalties can substantially affect the quality of the alignment.

Iterative Refinement [13,29] can be combined with progressive alignment approach in an attempt to correct the problem of having chosen wrong initial alignments. Every sequence from the final multiple sequence alignment generated by progressive alignment approach is removed and realigned repeatedly until the alignment score no longer improves. CLUSTAL W [17] is a

popular alignment program which extensively used the progressive alignment technique.

CHAPTER 2

PREVIOUS WORK

2.1 Alignment editors and visualization tools

Alignment techniques discussed above perform well in constructing sequence alignments. However, these are computational methods which give an alignment that may lose out on biological significance. The common scenarios considered while computing the alignment are insertion and deletion mutations, matches and mismatches. Other evolutionary scenarios like rearrangements and duplications are yet to be completely accounted for. Alignments tools like Shuffle Lagan [21] have been recently designed to compute alignments in the presence of rearrangements in sequences, and this remains an active area of research [10].

The dynamic programming method guarantees a mathematically optimal alignment, but this by itself does not guarantee biological correctness. The accuracy of the final alignment is largely affected by the scoring scheme adopted. Progressive methods base the multiple sequence alignment on an evolutionary tree; however the tree is often only an approximate prediction. If the sequences are well related, a correct tree is likely to be reconstructed, but this is generally not the case for distantly related sequences. The choice of wrong initial alignments can be corrected by integrating progressive with iterative refinement steps; however they still remain both computationally and biologically approximate.

It is also difficult to validate the correctness of sequence alignments by comparing them against benchmark databases. This is sometimes due to the over representation of some sequence families, thus invalidating the assumption that all alignments are independent of one another [24], as well as other reasons. In particular, good benchmarks are generally not available for DNA sequence alignments.

Biologically meaningful alignments are essential to make important discoveries. At present many biologists are forced to do manual adjustments to the automatically constructed alignment. Given the huge size of the underlying files, manual adjustments are extremely difficult. They include the placement of sites of experimentally confirmed homology or characteristic structural features in conformation, in cases when such similarities are not well reflected in the sequences, which may have misguided the automated (mathematical) optimization process. To make this process easier tools with user friendly interfaces are needed, such as GUI tools which will allow easy editing/adjustment of alignments. Editors and visualization tools are created with this issue in mind. The primary focus of this document is on editors, for which alignments are supplied as input. Many popular input formats are supported. Some of them are described below:

FASTA:

FASTA is the most popular format for representing sequences.

Description:

Every sequence in a FASTA file starts with a header which begins with the symbol “>”. The optional description of the sequence follows the “>” symbol. It generally includes the species name, information about the sequence, and its length.

The actual sequence starts in the line after the header and may extend over multiple lines. However in the FASTA alignment format, each block (which starts with a separate header) has to have the same number of characters, including gaps. The detailed description of the FASTA format can be found at <http://www.ncbi.nlm.nih.gov/blast/fasta.shtml>.

CLUSTAL W/ALN:

This format was introduced in the Clustal W alignment program. The file starts with the word “CLUSTAL” followed by the information about the version of Clustal W used. Every sequence starts with a header containing the sequence description. The actual sequence starts in the next line, and continues over as many lines as necessary. Every line can contain up to 60 residues. The total number of residues written so far is shown at the end of each line. The information about the rest of the sequences follows the same pattern. Each block of information ends with a line showing the level of conservation, according to the following scheme:

- "*" means that the residues or nucleotides in that column are identical in all sequences in the alignment.
- ":" means that conserved substitutions have been observed.
- "." means that semi-conserved substitutions are observed.

Example of a clustal/aln file taken from EBI (European Bioinformatics Institute) website, http://www.ebi.ac.uk/help/formats_frame.html.

PHYLIP:

Phylip files feature in two alternate formats, interleaved and sequential.

In both the formats, the alignment file starts with a line giving the number of sequences and the alignment length, both separated by blanks.

1. Interleaved format

The information about each sequence follows the start line. The sequence begins with a 10-letter wide description, followed by the sequence itself. The names should be filled to the 10 character width by blanks, when the name length is shorter than 10. Any printable ASCII/ISO character is allowed in the name, except for parentheses ("(" and ")"), square brackets ("[" and "]"), colon (":"), semicolon (";") and comma (","). The sequence itself is written in groups of 10, separated by one or more spaces. The block of sequences ends with the last sequence in the alignment. If more residues are to be written a new block is started without including the names of the sequences in it. There may or may not be empty blank lines between the blocks.

An example of an interleaved phylip file can be found at the PHYLIP website:

<http://evolution.genetics.washington.edu/phylip/doc/main.html#inputfiles>

2. Sequential format

For sequence descriptions, the sequential format follows the same convention as interleaved. Each sequence is written completely in groups of 10 over multiple lines.

An example of a sequential phylip file can be found at <http://evolution.genetics.washington.edu/phylip/doc/sequence.html>

MASE

The beginning of a MASE formatted file must contain a header containing at least one line (but the content of this header may be empty). The header lines must begin with `;'. Every sequence must start with one (or more) comment line. Comment lines begin with the symbol the character `;' and they may be empty. The name of the sequence follows in a separate line after the last comment. The sequence itself is written on from the next line.

An example of a mase file can be found at SeaView website, 27. <http://pbil.univ-lyon1.fr/help/formats.html>

2.2 Available tools

There is a variety of alignment visualization and annotation tools available, and also several sequence editors. Artemis [5], Genquire [31], PipTools [12] are some of the visualization tools available. We shall here describe several major ones.

2.2.1 VISTA

The VISTA visualization tool [19] has been developed and hosted at Genomics Division of The Lawrence Berkeley National Laboratory.

It provides two ways of visual representation of an alignment: Vista Browser and Vista Track.

Vista Browser allows the user to view pre-computed whole genome alignments of many species. When new genome assemblies are released they are aligned to previously stored sequences and made available to the community. Vista browser is a Java applet and Java 2 is required to view the alignment. To browse a whole genome alignment, a base genome should be selected and a Refseq (a database maintained by NCBI, National Center for Biotechnology Information, that aims to provide a comprehensive and non-redundant set of sequences, including genomic DNA, RNA and protein products, for major research organisms) gene name or a position range in the genome to be viewed must be specified.

Conserved regions are calculated as segments of certain length n with at least m percent identity with respect to the base sequence. These parameters can be set by the user. The degree of conservation within discrete windows in the sequence is plotted along the Y-axis of the preview area, and the plotted points are connected by a curve, with multiple view areas stacked on the top of each other as illustrated in figure 2.1. Conserved regions are highlighted under the curve, with the conserved non – coding regions colored in red and conserved exons colored in blue. A legend is provided that explains how different regions are represented.

Vista track, accessible through the Vista Browser displays results of comparative analysis in the context of the annotations of whole human genome. It dynamically creates VISTA plots for each defined region.

Vista Browser and Vista Track are linked to the Text Browser, which allows users to examine information about each sequence aligned to the selected region on the base genome.

Vista can be accessed at <http://genome.lbl.gov/vista/index.shtml>. A new visualization tool PhyloVista [2] has been developed by the Vista team.



Figure 2.1 A screen shot of the Vista Browser

2.2.2 CINEMA

CINEMA (Color Interactive Editor for Multiple Alignments) [23] is an editor for manipulating multiple sequence alignments and sequences. It is implemented as a Java applet.

To overcome the security restrictions imposed on accessing the local file system, on Java applets, the developers of CINEMA have set up a server that can save user data, which can then be manipulated to send back the results to the user either through the browser or email. The server has been implemented using CGI and Perl scripts. The CINEMA editor accepts input in many formats .It takes an alignment from

the local machine or from the PRINTS database system maintained by the server. The output is mailed back, or presented in a GIF/text format.

The alignment editing features include cut/copy/paste of sequences, and insertion and deletion of gaps. Sequences can be combined and edited as a group. Amino acids are colored based on their properties and users can specify their choice of colors. Individual sequences can be manipulated and new sequences can be appended to the alignment window using the sequence editor.

The CINEMA application comes with a motif editor. The motifs are automatically uploaded when selected.

This application also provides a novel way of viewing alignments by introducing split panes. Alignment is shown in multiple panes; and multiple regions of the alignment can be examined simultaneously. Navigation within each pane is controlled separately, which enables independent visualization and comparison.

Optional extensions called pluglets are available with the editor. The skeleton pluglet is a 3D backbone viewer of protein sequences. Clicking on a sequence within the alignment sends its structural information to the 3D viewer, provided that one is available. The 3D viewer displays the structure of the protein, which can then be zoomed in and out, rotated and spinned.

The 6F pluglet translates a given DNA sequence in its six reading frames and displays the result to the user.

CINEMA editor can be accessed at:

<http://www.biochem.ucl.ac.uk/bsm/dbbrowser/CINEMA2.02/index2.html>

Instructions on usage are given at the website.

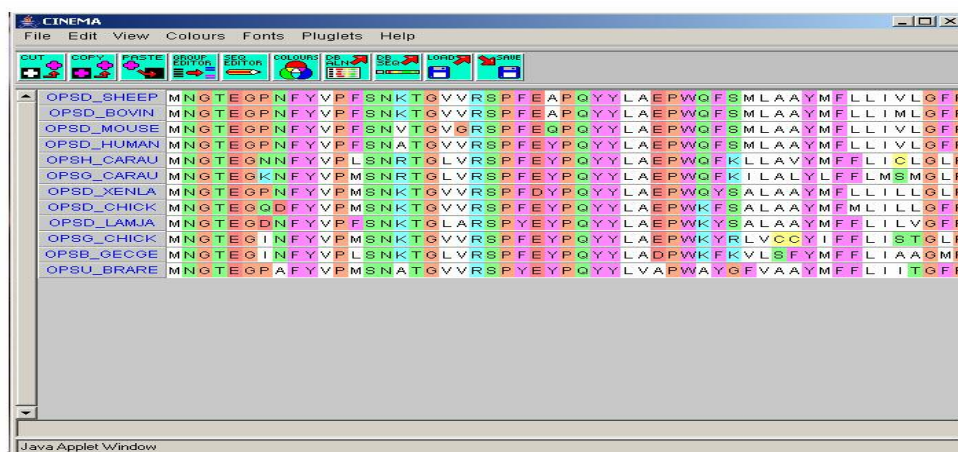


Figure 2.2 A screen shot of the CINEMA viewer

2.2.3 SeaView

SeaView [14] is a multiple sequence alignment editor developed at Université Claude Bernard – Lyon, France. It is implemented in ANSI C.

The editor takes an alignment or sequences as input, in various formats. Developers of SeaView have devised their own alignment format called “MASE” which has been described earlier in the document.

The sequences can be saved in a variety of formats including PDF. SeaView interfaces with two alignment programs CLUSTAL W and MUSCLE [18] to align selected sites from different selected sequences. Only equal length sites can be aligned, the remaining portions of the alignment/sequences are discarded. Any alignment program can be used to handle these sites, CLUSTAL W being the default. The editor provides the options to include comments and footers below the sequences. Coloring schemes are available to highlight important residues.

Other editing features like cutting and pasting of residues, insertion and deletion of gaps, duplication and pasting and deletion of sequences are also provided. Sequences can be grouped and saved as a set. Search options for finding patterns and navigating to a particular base in the sequence are the other notable features of this editor.

The application can also be launched from command line, instructions given on the web site.

SeaView can be obtained from <http://pbil.univ-lyon1.fr/software/seaview.html>.

Instructions for the use of the editor are given at the website.

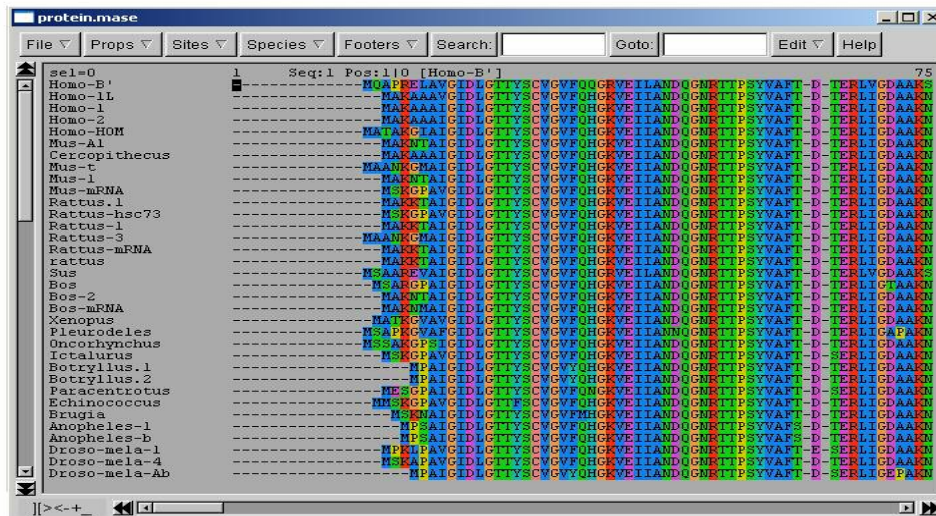


Figure 2.3 A screen shot of the SeaView editor

2.2.4 Jalview

Jalview [6] is a multiple alignment editor written in Java.

It uses files which can be input from the local machine, or from a URL, or supplied by pasting the sequences in an input textbox. Similarly files can be saved in any one of the formats supported.

Jalview offers many features to manipulate individual sequences. These include cut/copy/paste of residues, selection/deselection of residues, hiding sequences and columns, moving the selected regions of alignment to a new position and the insertion/deletion of gaps. Different color schemes are supported for protein sequence alignments. The coloring is intended to represent the chemical properties of the constituent amino acids. When specific sequences are selected, the colors are applied only to the selected environments. User defined color schemes are also supported.

Sequences can be categorized under different groups identified by a group name and each group can be assigned a different coloring scheme. Regions of alignment can be manipulated through cut/copy/delete operations. Jalview interfaces with utilities, such as alignment of sequences using Clustal W [17] or MAFFT [18] multiple sequence alignment tools, prediction of secondary structure of proteins based on the alignment, using the Jnet tool [7]. Jalview integrates the Jmol [31] 3D viewer utility to visualize a selected protein sequence. Pairwise alignment of any two selected sequences and of the calculation of phylogenetic tree from the alignment, using different techniques, are the other features offered. Annotations such as consensus residue of every column, conservation of physical/chemical properties in every column are calculated and displayed beneath the alignment window. Different functional sites in sequences are colored based on information supplied through annotation files. Jalview comes in two variants: application and applet. The main application allows a connection to multiple web services, as well as additional features such as printing the alignment and

annotating the alignment. It needs to be downloaded and installed on the user's machine.

The applet version runs in web browsers and provides a useful interactive display for alignments, features and annotations files. However, it does not have the full functionality of the main application, such as saving files or, running web service jobs due to security restrictions imposed on applets.

The Jalview editor and instructions can be obtained from <http://www.jalview.org/>

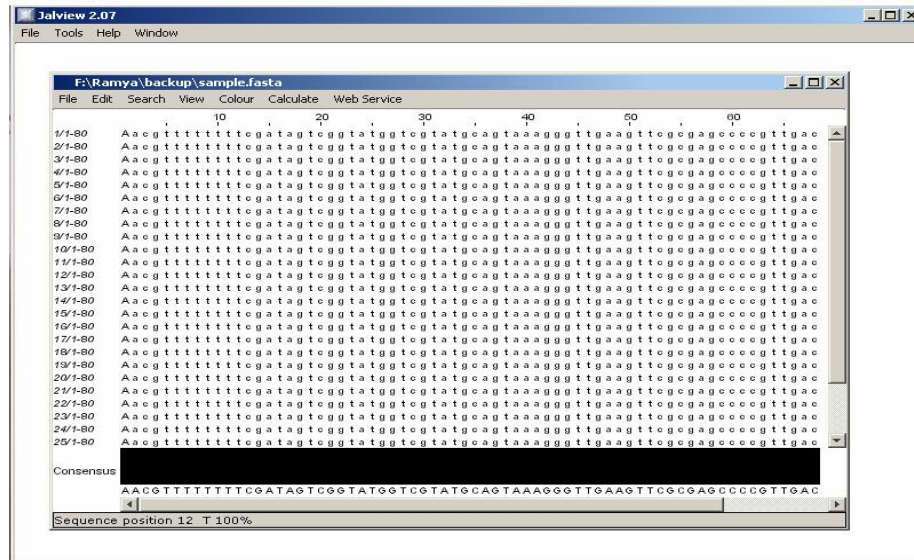


Figure 2.4 A screen shot of the Jalview editor

CHAPTER 3

SEQUENCE ALIGNMENT EDITOR

3.1 Summary

The editors discussed above offer good visualization features, and some of them integrate a host of utilities across the web. However, the functionalities provided are suitable for manipulating the sequences making the alignment, rather than the alignment as a whole.

Alignment editing can be defined as manipulating the blocks where the order of residues in the individual sequences should not be disturbed. The applications seen so far are prominently sequence editors. With this issue in mind, we have undertaken the development of a tool to assist biologists in modifying the alignments by placing experimentally verified functional areas under conformation, which are at times missed by the alignment building software.

Our Sequence Alignment Editor is a user friendly stand alone utility. It has been developed using Java Swing package.

3.2 Implementation platform

Swing is a tool kit supplementing the Java language in order to facilitate the developing of standalone GUI applications. It is a part of the Java Foundation Classes (JFC) and it is based on the Java AWT GUI tool kit. Unlike AWT, Swing components are lightweight components. For example, every Swing component paints its rendition

on the graphic device in response to a call to a Java method called `paint()`. However unlike AWT components which delegate the painting to their OS-native "heavy weight" widget, Swing components are responsible for their own rendering. It gives a consistent look across all platforms with a platform independent look and feel known as the "default look and feel" or the "metal look and feel".

Swing also comes up with a native operating system look and feel if one is available; otherwise it settles down for the default. With a rich set of user interfaces and interactivity added to the Java application and assured portability, Swing was the obvious choice to implement our tool.

3.3 Editor layout

The editor takes an alignment file as input. Once loaded, the alignment is displayed on an alignment window, which provides a number of informational fields for the description of the input file as well as the navigation tools. For details, please see the instruction manual, provided in APPENDIX A. A screen shot of the alignment layout is given in figure 3.1.

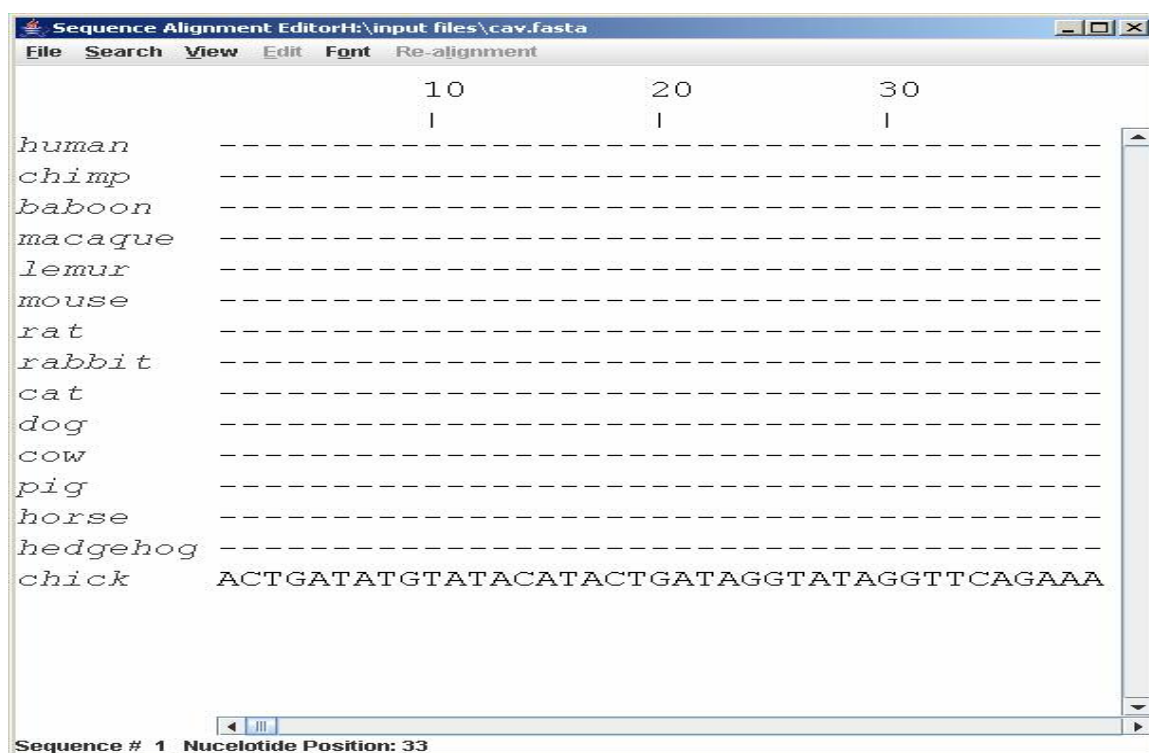


Figure 3.1 Screen shot of editor layout

3.4 Display of annotated regions

In addition to editing features, the editor also offers visualization of annotated regions in the sequences. The annotations consist of the information about biologically important, either as computationally predicted or experimentally verified regions in the sequences. For example, the start and end position of a gene, exon boundaries and 5' and 3' locations make the annotations for that gene. Annotations can be supplied to the editor as an annotation file, complying with our custom designed format.

Annotated regions are represented using different colors and a legend is provided to show the specific colors used for specific features. The highlighting of the annotated regions has been designed to give the user a good idea where those areas have

been placed in the alignment. An illustration of the color scheme legend is shown in figure 3.2.

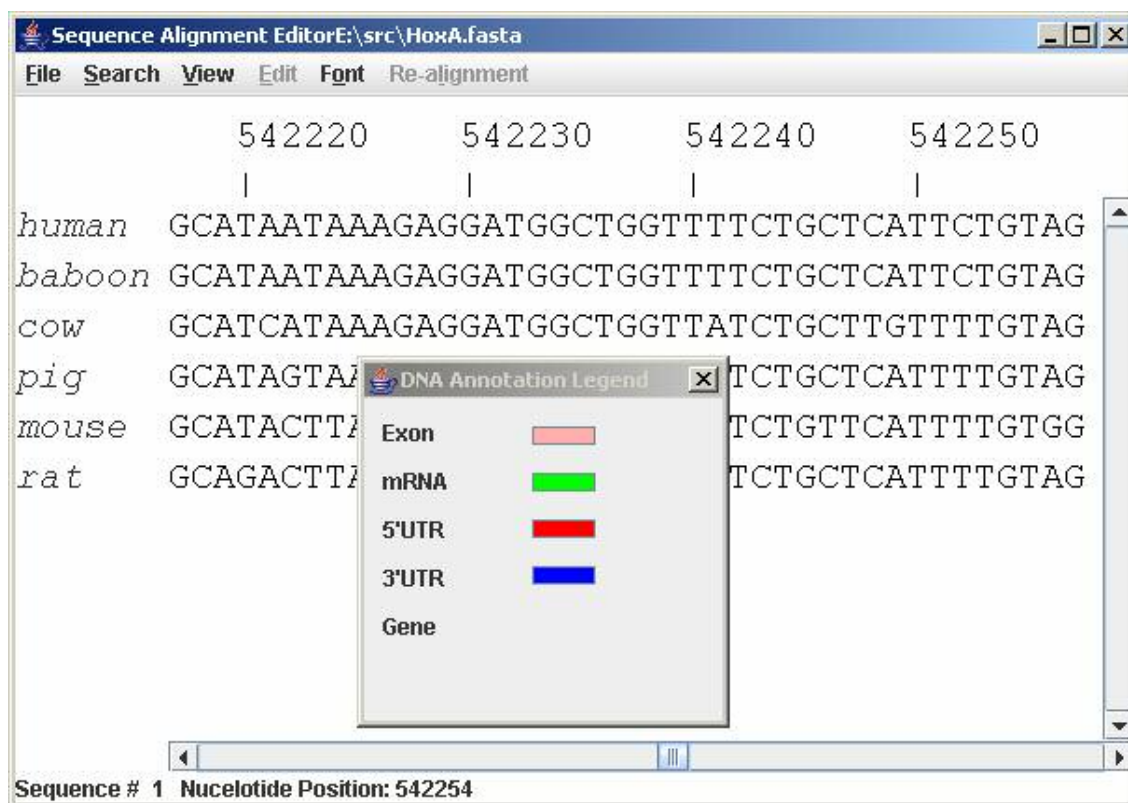


Figure 3.2 Annotation legend

3.5 Alignment editing

In order to post process an automatically constructed alignment, the region which requires editing needs to be marked or frozen. Its boundaries indicate that only the marked area needs to be corrected and that rest of the alignment should not be disturbed. The marked area is referred to as the region under editing. Once this region has been locked, experimentally verified sites or sites of functional significance that ought to be under conformation can be dragged together. The dragging of sites results in the bases getting displaced from their original positions in the current alignment,

resulting in gaps at the beginning and end of the region under editing. This process is illustrated in figures 3.3 and 3.4.

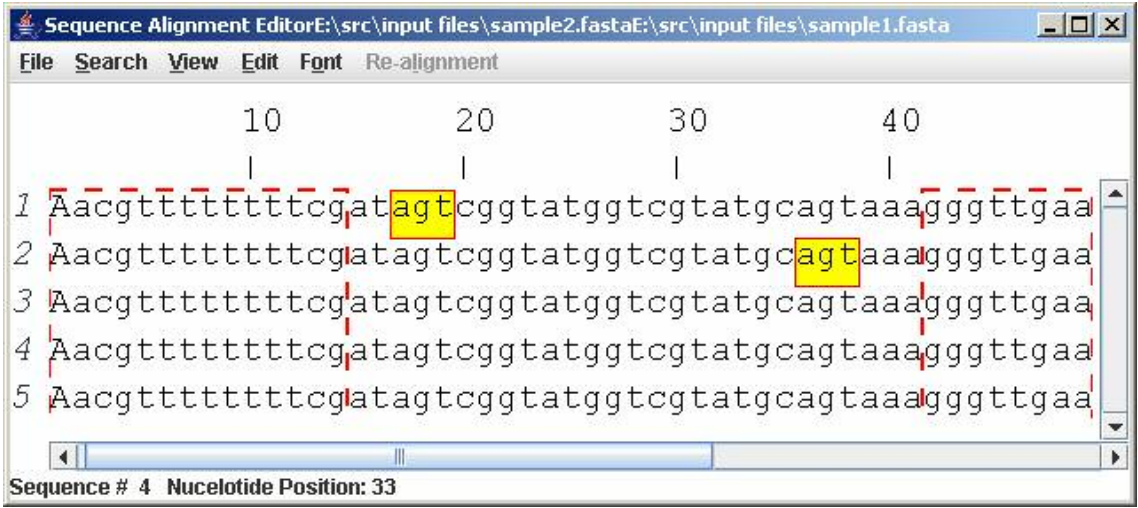


Figure 3.3 Selection of sites in sequences 1 and 2

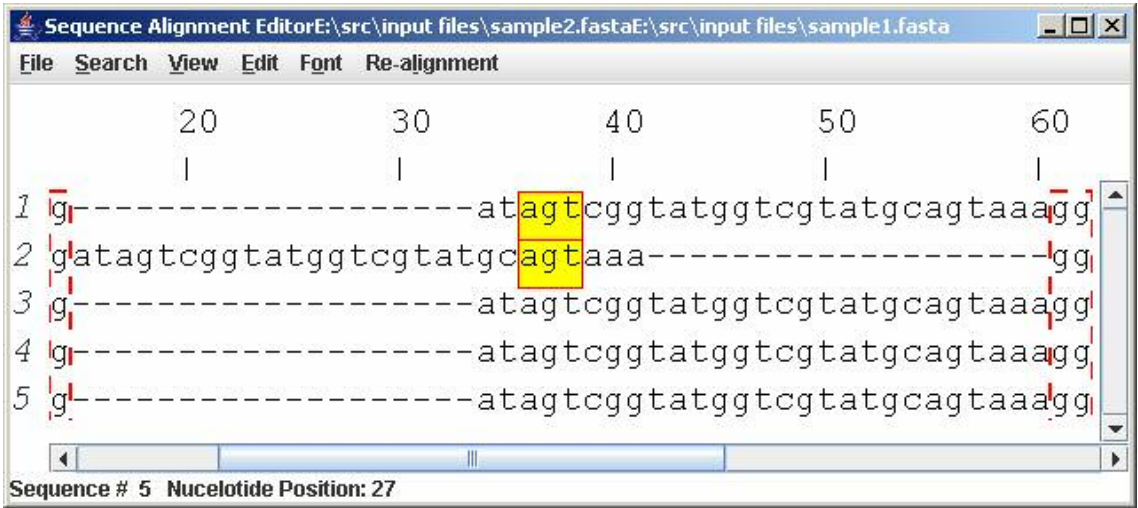


Figure 3.4 Placement of sites in one column

Gaps can also be introduced within the newly created area of conformation, if the dragged sites are not of equal length. As illustrated in figure 3.5, the editor has functionality to efficiently deal with this type of situation.

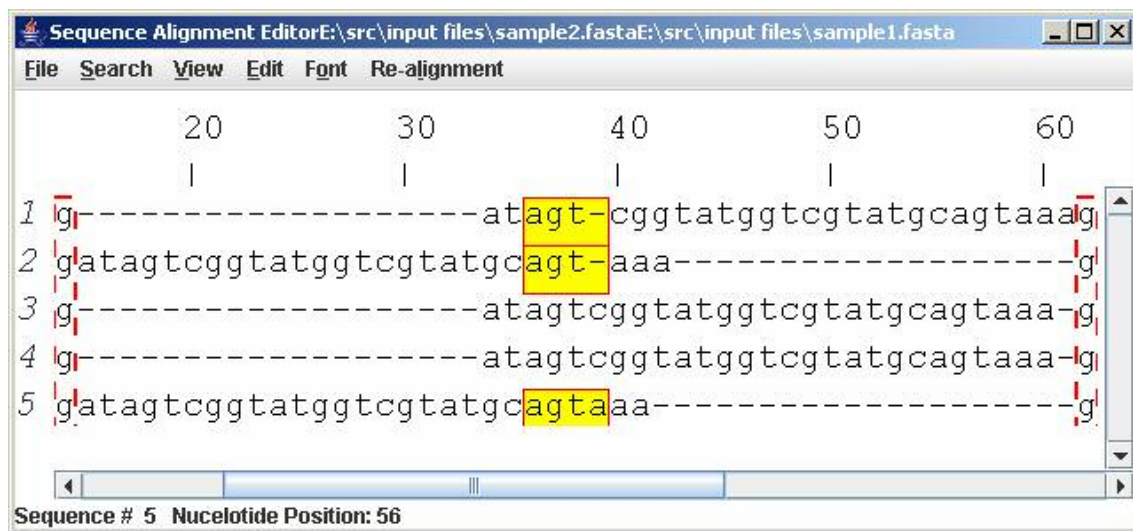


Figure 3.5 Gaps within the area of conformation

Once the user has finalized the decision about the placement of sites, the modified region is realigned. The motivation for realigning is to ensure that sites are properly conforming with one another, which can provide further insights. In certain cases, the user may have intended to have only the start of each site placed in conformation. This may be because information is available only for the start of the site and not about the rest, in which case the length of the areas brought under conformation is not known and the best way to gather further knowledge would be to realign and analyze them. Along with the target section, the disturbed areas flanking it are also realigned. This returns the disturbed regions in alignment and removes most of the artificially introduced gaps. The boundaries of the disturbed regions for realignment are determined based on the alignment quality. The boundaries are just not taken as the ends of the region under editing. The alignment is scanned for columns of good conservation and when such columns are found, the boundaries are fixed there. This automatic anchoring is done when the user is not sure about the boundaries of the

region under editing and the extent of the disturbed region to be realigned. The disturbed region to the left of the edited area has its right end fixed at the position just before the area start .Its left end is determined based on the concept of information content. Information content is a measure of how well a region of the alignment is conserved, implying its significance, and, consequently disturbing such sites should be avoided. For a region of alignment called site, to be considered important, it must carry enough information to indicate its significance. The information thus carried is known as information content of the site and is calculated based on the information theory [26]. The formula [25] for calculating information content is:

$$R_{\text{sequence}}(\text{site}) = \sum_{i=1}^L \sum_{b=A} f(b,i)/M \log_2(f(b,i) * N/nb * M)$$

where

i denotes a column in the region under consideration

L = total number of columns in the region

$f(b,i)$ = frequency of a nucleotide in column i .

M = total number of sequences in the alignment.

nb/N = general frequency of occurrence of a nucleotide in the genome

Alignment column triplets are scanned starting from just before the start of the disturbed left region, and their information content is measured. The scanning stops when a minimum score of 5 bits of information is reached, or when the number of bases scanned reaches 1000. Columns with gaps are ignored during the scanning process except for their contribution to the total count. The last scanned column is fixed as the

left end of the right disturbed region. The same procedure is then repeated to determine the boundaries of the right disturbed region. Once their boundaries are fixed, both the regions are realigned.

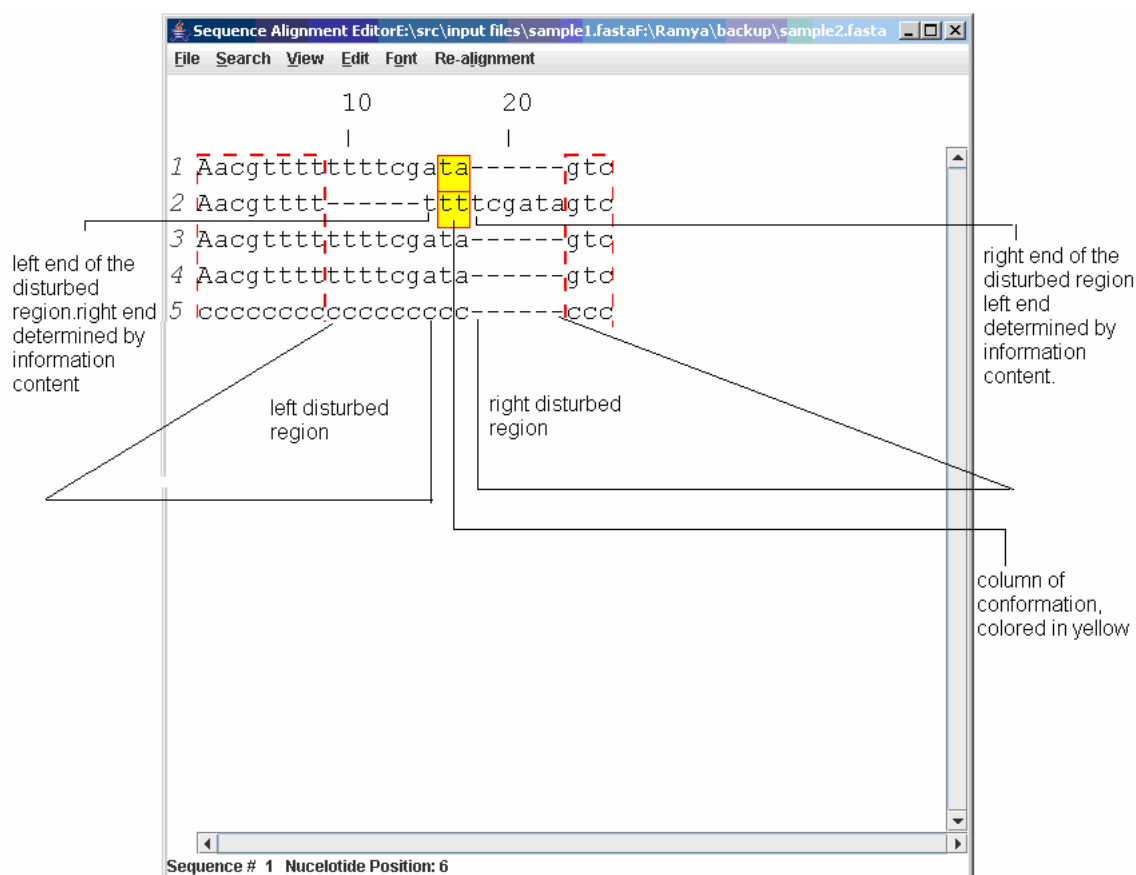


Figure 3.6 Screen shot showing the area of conformation and disturbed regions

The realignment is carried out by interfacing the editor with the CLUSTAL W, version 1.83 alignment program taken from the EBI (European Bioinformatics Institute) website, <http://www.ebi.ac.uk/clustal w/>

The three realigned sections are stitched back into the undisturbed alignment, and the layout reflects the conformation of sites. The user can continue editing by selecting a new region that requires manipulation, and walk through the same steps described above.

The editing of the alignment can be continued for as long as the user wants. However, at any given time only one region can be edited.

CHAPTER 4

EXAMPLES

4.1 Input

The editor was tested with a sample input alignment of Hox A regions of human, baboon, cow, pig mouse and rat. The alignment was 1,031,804 bases long. Basic annotations for the human Hox A region were supplied to the editor.

4.2 Hox genes

Hox genes contain a homeobox pattern around 180 base pairs long that codes for an active domain, which forms a part of a transcription factor important for a vertebrate development. The proteins coded for by homeobox genes are called homeodomain proteins. Hox genes are responsible for determining the positioning of certain body regions such as limbs in a developing fetus. They are well conserved across vertebrate genomes indicating that they have been shielded from mutations due to their significant role they play in the development of an organism [15]. Hox genes in humans are grouped in Hox A, B, C and D clusters, each cluster containing several genes. The human Hox A cluster is located on Chromosome 7.

4.3 Alignment before editing

The alignment has been analyzed at hoxA11 gene, by navigating to the start of the exon that is defined for hoxA11 gene as shown in figure 4.1.

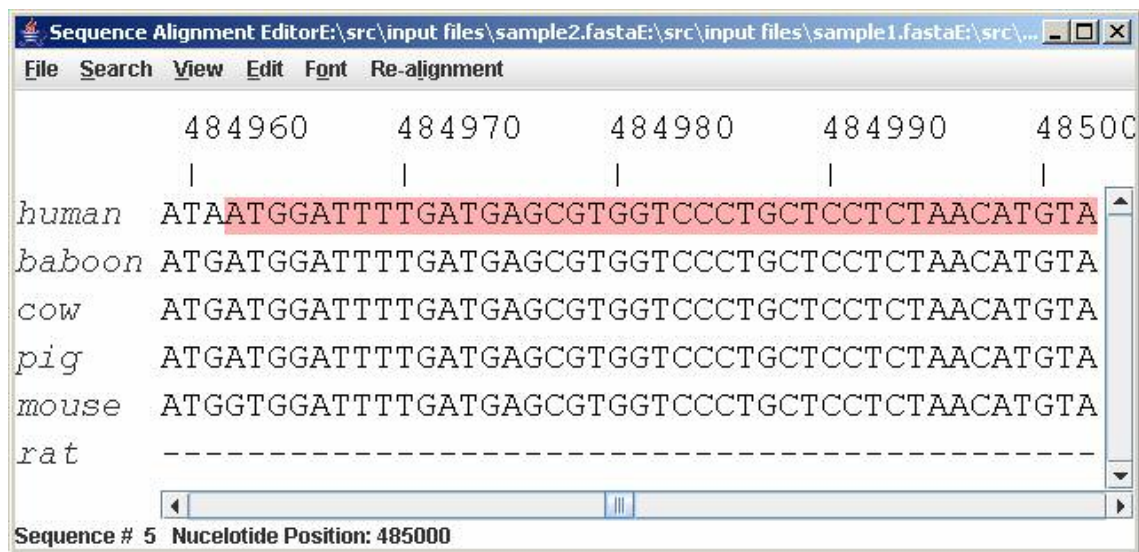


Figure 4.1 Screen shot of the Hox A input alignment, at the beginning of exon1 of HoxA11 gene, in the human sequence

The triplet “ATG” starting the area highlighted in pink in the human sequence denotes the start codon (the 3-base region which denotes the start position in a gene that is responsible for producing a protein [15]) of the hox A11 gene, in the human species(highlighted area denotes the whole exon). From figure 4.1 it can be seen that the triplet “ATG” has been marshaled in all the sequences except mouse and rat. The gaps in the rat sequence show that there are no corresponding bases in the rat sequence, most likely because that particular segment in rat was not available at the time this alignment was built.

In figure 4.1, “ATG” triplet in mouse sequence can be seen starting at position 484972. Using the Sequence Alignment Editor, the “ATG” triplet starting at position 484972 has been selected and dragged under the already marshaled “ATG” column, as illustrated in figures 4.2 and 4.3.

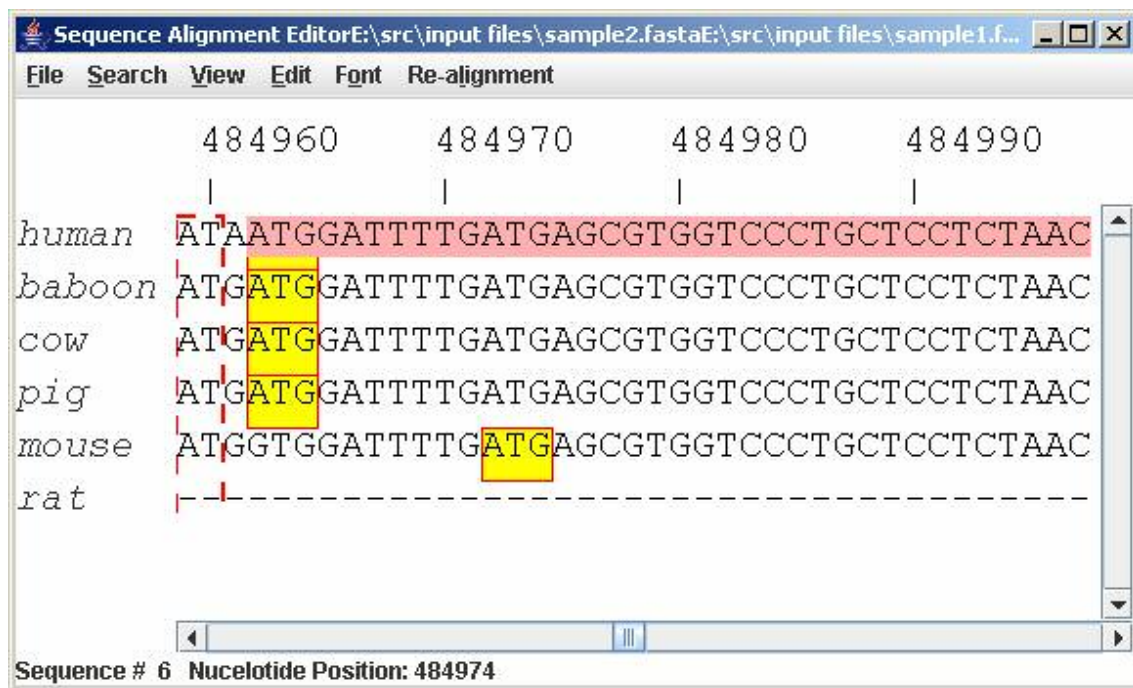


Figure 4.2 A screen shot showing selection of “ATG” triplet in mouse sequence

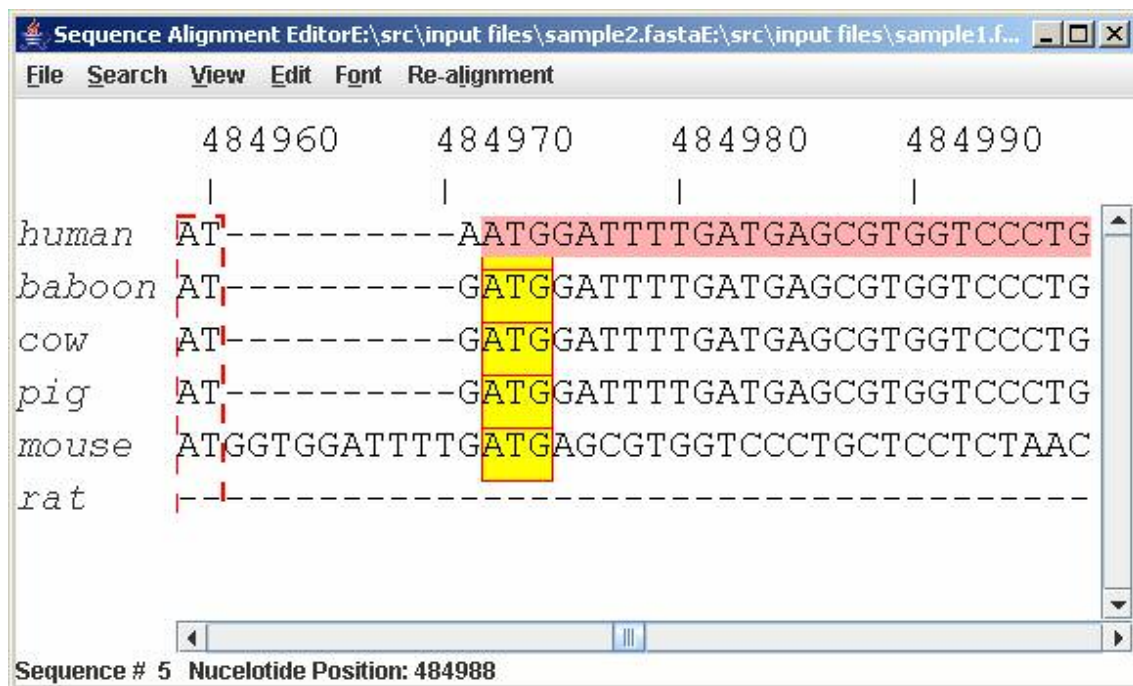


Figure 4.3 A screen shot after placing the “ATG” triplet of mouse

in conformation with the “ATG” triplet of other sequences

After fixing the column of conformation the disturbed regions on either side of the yellow coloured column in figure 4.2, along with the conserved column are realigned by using the Clustal W alignment program. The realigned regions were patched with the undisturbed regions of the original alignment in order to give the final post processed version.

4.4 Alignment after post processing



Figure 4.4 Hox A Alignment after post processing

Figure 4.4 shows the “ATG” triplets are lined up. The conserved column is shown in green.

CHAPTER 5

FUTURE WORK

The sequence alignment editor provides good editing features to post process sequence alignments, the most important being the placement of important sites under conformation followed by a realignment of the disturbed regions and column of conservation. Having supported this main feature along with other visualization and editing features described previously, the editor can be further enhanced by supporting more features.

Some of the immediate extensions to the editor are described below:

Page fault support:

The editor currently supports alignment files up to 25 MB. Genome alignments are huge and in the future as more species are discovered and sequenced, the construction of long and huge alignments will become an inevitable task in bioinformatics. To accommodate such long alignments, the concept of page fault will be incorporated in the alignment.

Printing facilities

Currently the editor supports features to save an alignment in any desired format supported by the editor. Printing features to present the alignment in desired layout such as text or pdf or post script will be incorporated.

Multiple alignment program support

The editor interfaces with Clustal W alignment program to perform the realignment of sections of the edited alignment. In the future, it will be interfaced with different multiple alignment programs such as MLAGAN [20], MAVID [4] and the user can choose an alignment program to use with the editor.

Automating Annotations

Currently editor takes annotated files given by the user as input. In the future the editor will interface with databases across web, which provides annotations. Examples of databases having annotations are GenBank, SwissProt.

Swiss-Prot is a manually curated biological database of protein sequences. It is maintained by the Swiss Institute of Bioinformatics as one of the constituent database of UniProt, Universal Protein Consortium.

Specifics for protein alignments

Sequence alignment editor can currently display protein alignments and highlight its annotated regions. The specifics of fixing of boundaries for the disturbed regions have to be determined, based on the different substitution matrices available for protein sequences.

Simultaneous display of realignment

Simultaneously display the alignments of same regions that are constructed by different alignment tools.

APPENDIX A

INSTRUCTION MANUAL

The alignment that is given as input to the editor is first validated to check if it conforms to a supported file format. If the validation succeeds, the alignment is loaded to the editor, otherwise the editor throws up an error message to the user informing about the invalid format and also the formats supported by the editor.

The file name along with its path is displayed in the title bar of the editor. Each sequence is displayed in a single line with the sequence description displayed to the left of the sequence. The alignment window is supplied with scroll bars to enable the accessing of different regions of alignment. The alignment window has a ruler that marks every 10th position of the alignment. In addition, a status bar at the bottom of the alignment window shows the current sequence and base corresponding to the position of the mouse.

The editor offers all its functionalities, packed under 6 menus, the menus being:

- File
- Search
- View
- Edit
- Font
- Realignment

NOTE: The alignment shown in the screen shots is artificial.

1. File Menu

This menu is the gateway to the application. The keyboard shortcut to this menu is ALT+F. It provides the following alignment file handling functionalities:

1.1. Open

The “Open” menu item allows the user to load an alignment file into the editor. The keyboard shortcut is Ctrl+O.

Types of alignment files supported

FASTA (.fasta)

CLUSTAL W (.aln)

MASE (.mase)

PHYLIP (.phylip)

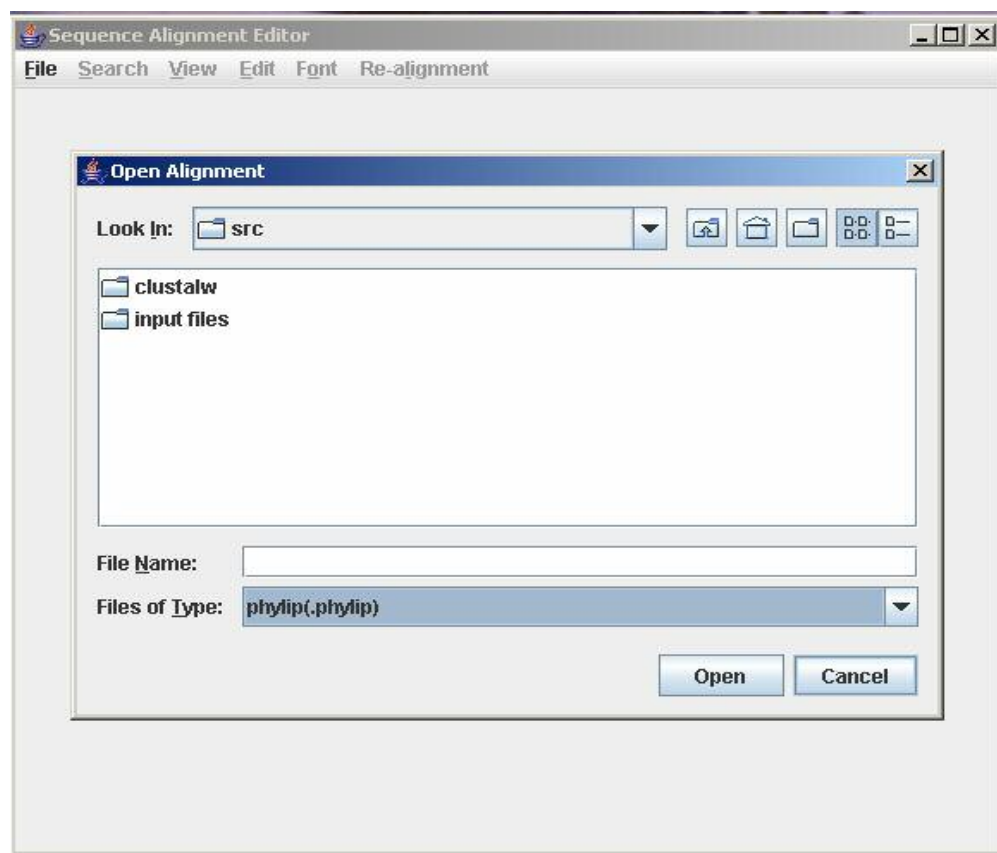


Figure A1 File formats supported by Sequence Alignment Editor

1.2. Open Annotation

This menu item loads the annotation file for the alignment file (opened in the editor) to the editor. The keyboard shortcut is Ctrl+A. Annotation files should conform to the format described below:

Every annotation file should start with the line “TYPE DNA”, indicating that the annotation is for a DNA alignment. This line is for extensibility in which case it can be specified as “TYPE PROTEIN” for a protein alignment.

Annotations for each sequence should follow the start line, beginning with a sequence header. The sequence header starts with the character “>” and contains

description about the sequence, in the same way it is specified in the alignment. The sequence description should not contain blanks and the individual words within the description must be separated by an underscore “_” symbol. The actual biological details start from the next line. For every sequence, gene should be described first as “gene” followed by gene description after a space. Information about other regions follows the gene description in respective separate lines. Each line contains the description of the region, the start and end locations of the region separated by blanks. The description of the region should not have spaces in it. The individual words within the description should be separated by underscore “_” symbol.

An artificial sample annotation file:

TYPE DNA

>human_gene_A

gene geneA

exon 40 50

exon 55 60

1.3. Save As

This menu item allows the user to save the alignment file either under the same name or under different name. The keyboard shortcut is Ctrl+S.

NOTE: The “Save” operation takes effect only if the user decides to do so. The file is not saved with the latest modifications, when the editor is closed. Similarly the current state of the alignment in the editor will not be reflected in the alignment file on disk.

1.4. Save

This option saves the alignment file under the current name. Keyboard shortcut is Shift+S.

1.5. Exit

This option closes the editor. Keyboard shortcut: is Ctrl+X

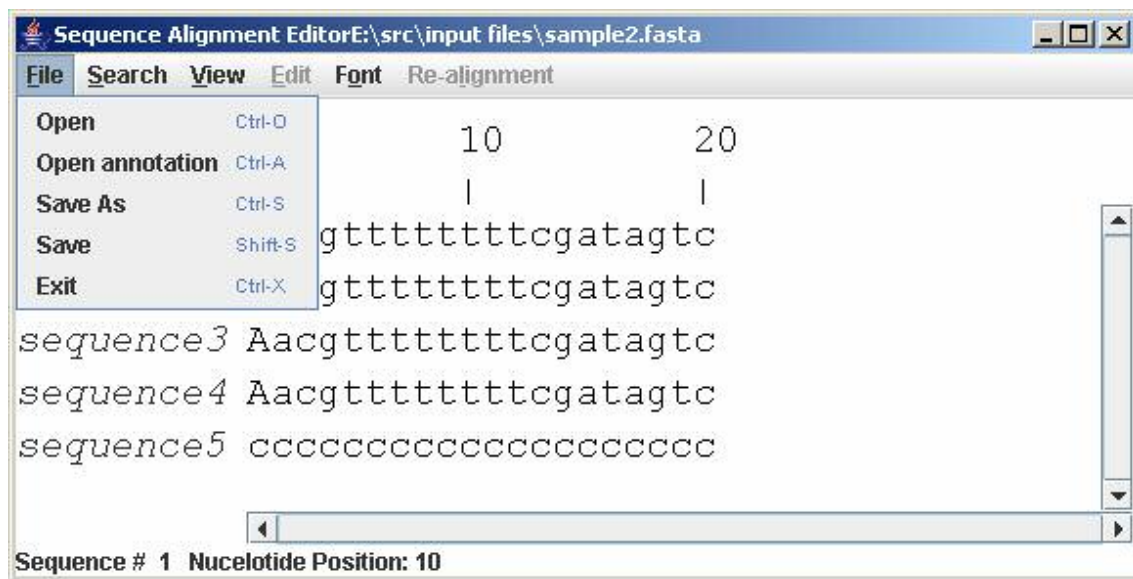


Figure A2 File menu

2. Search menu

This menu allows the user to search for desired bases in any region of the alignment. Keyboard shortcut to this menu is ALT+S.

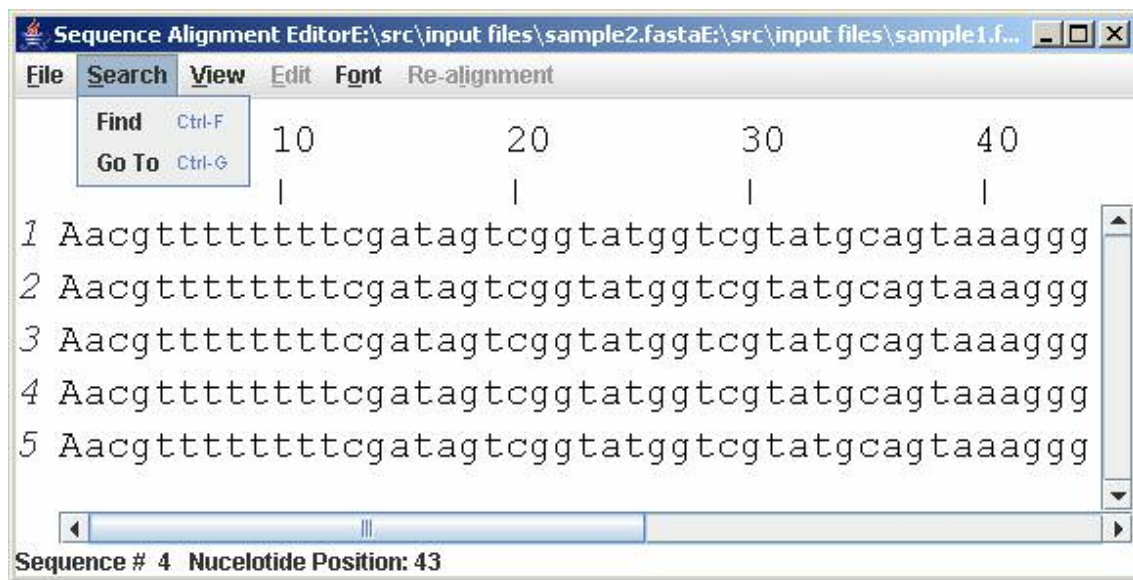


Figure A3 Search menu

The Options under this menu are:

2.1 Find

The Find option searches for the pattern to be found and highlights it. Keyboard shortcut is CTRL + F. User can specify the range of sequences as well as the range of bases to be searched.

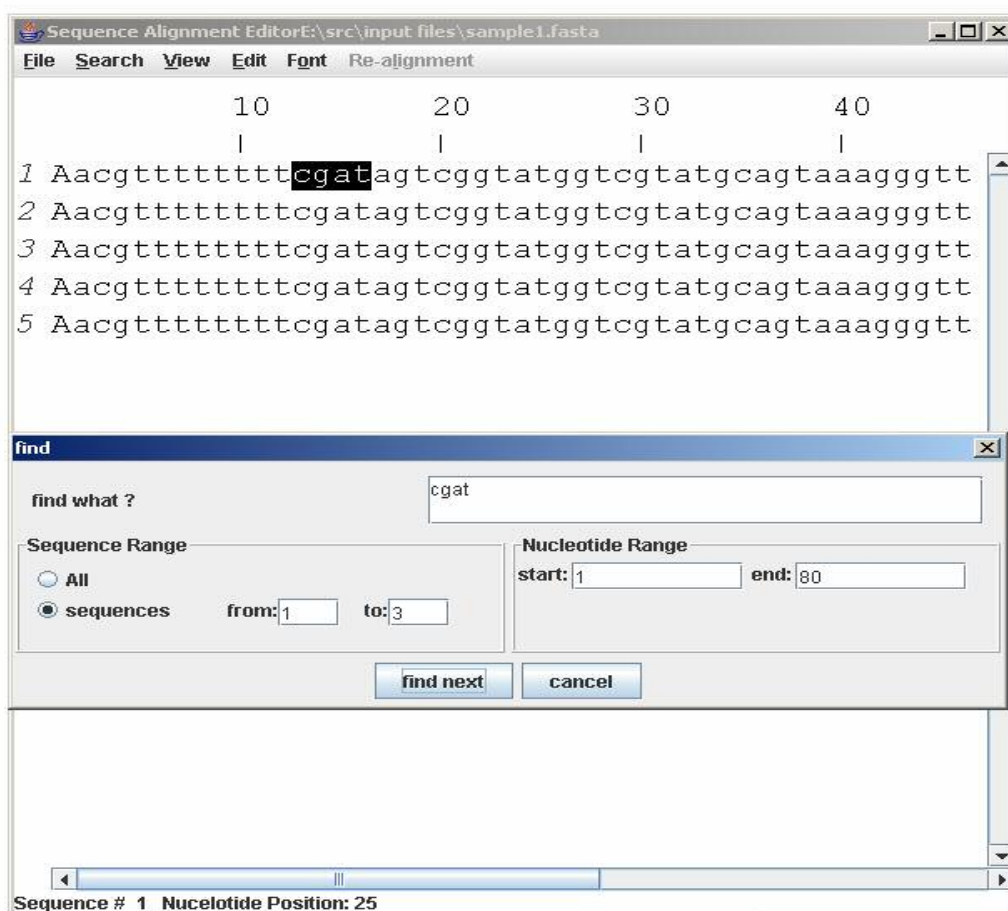


Figure A4 Find option

2.2 Go To

Go To option takes user to the desired location in the alignment depending on the sequence chosen by the user. Keyboard shortcut is CTRL + G. This also serves as a location reference guide for the selected sequence by displaying the total no. of bases in the sequence and the current position of the sequence.

Apart from being a part of the Search menu, Go To is also a part of a pop-up menu which gets displayed with a right click on the alignment area.

Options under Go To:

2.2.1Position

Go to the position asked by the user.

2.2.2Number of Bases to the Left of Current Position

Go to the left of the current position by the no. of bases specified.

2.2.3Number of Bases to the Right of Current Position

Go to the right of the current position by the no. of bases specified.

2.2.4Beginning of the Alignment

Go to the start of the alignment.

2.2.5End of the Alignment

Go to the end of the alignment.

2.2.6Annotated Region

Navigate, to the start or end of the selected annotated region.

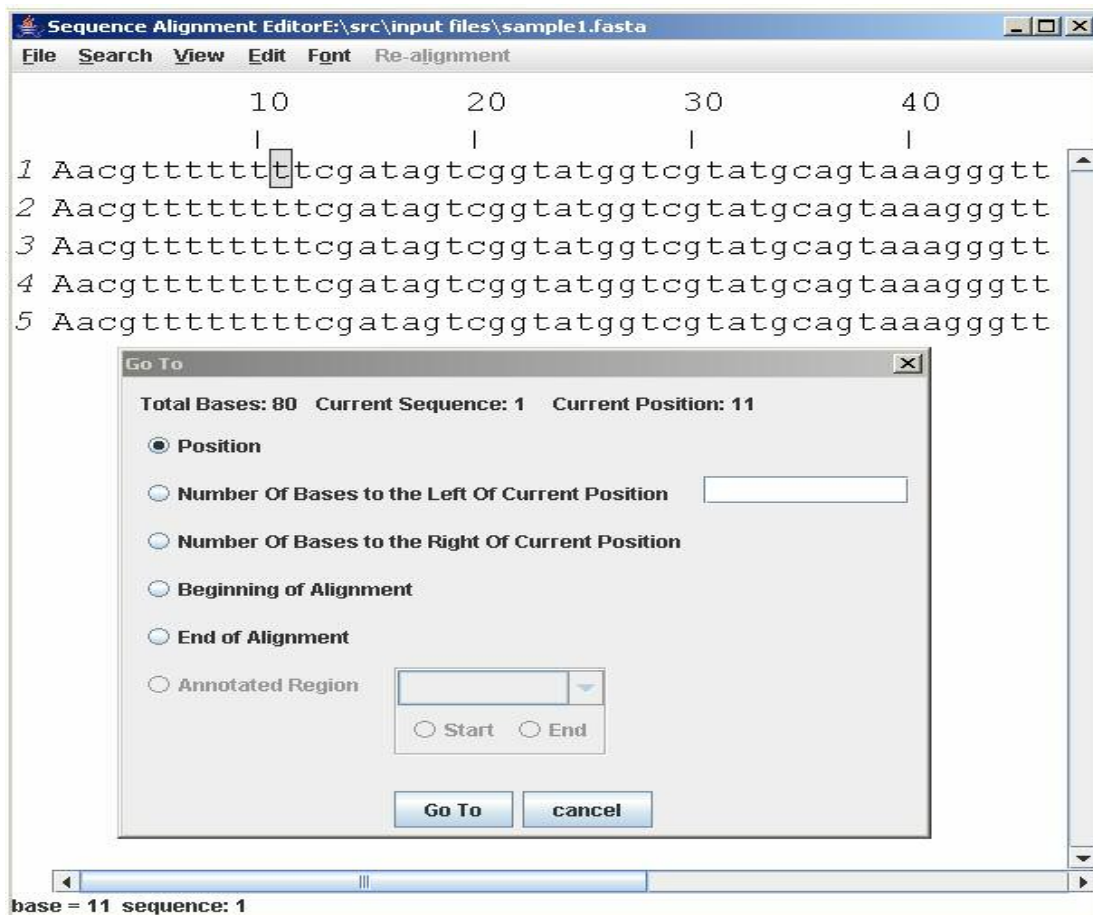


Figure A5 Go To option

3. View menu

This menu provides different ways to visualize the alignment. Keyboard shortcut for this menu is ALT +V.

Ways of visualization:

3.1 Bases in upper case

Entire alignment displayed in upper case letters

3.2 Bases in lower case

Entire alignment displayed in lower case letters.

3.3 Normal View

This is the un-annotated view where the annotated regions are not marked.

3.4 Annotated View

Annotated regions are highlighted under different hues.

3.5 DNA Annotation Legend

This is the legend of colors used for highlighting nucleic acid annotations.

3.6 Protein Annotation Legend

This is the legend of colors used for highlighting protein annotations

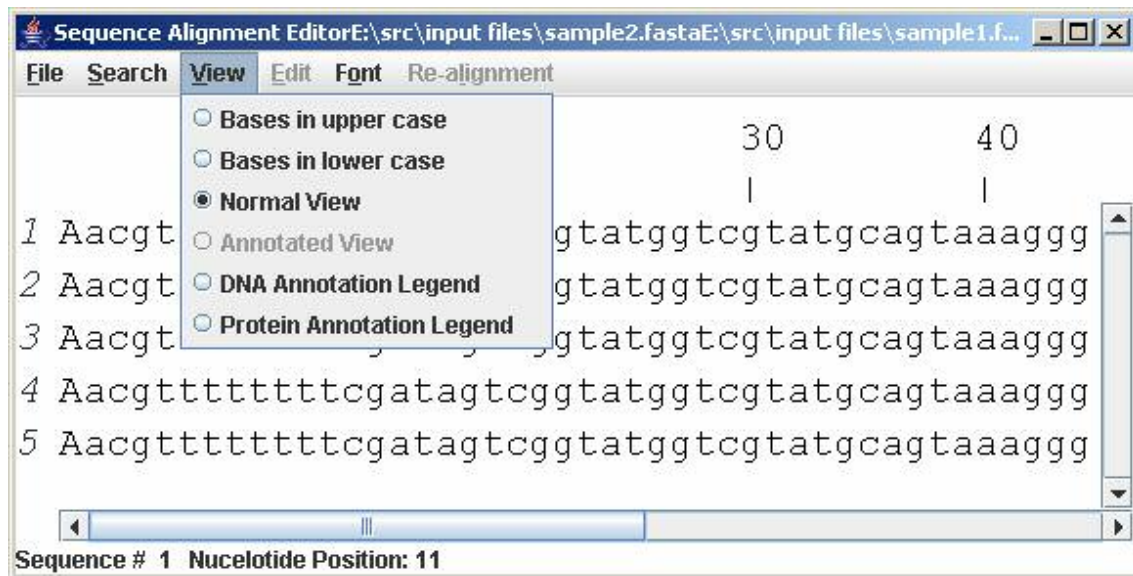


Figure A6 View menu

4. Edit Menu

This menu gives the core editing functionalities for alignment editing. Keyboard shortcut for this menu is ALT + E.

The editing features are grouped under 3 sub-menus.

4.1. Fix Region

This menu allows the user to finalize the alignment region selected for editing. The area meant to be edited must be frozen with the “**freeze**” option in order to be edited.

Options under Fix Region:

4.1.1. *Freeze*

“Freeze” fixes the boundaries of the alignment area that needs to be edited. Once frozen, the region of interest is fixed and the flanking areas marked with dashed “red lines”, indicating that those regions are not mean to be disturbed. Once the region of interest is fixed it can be changed only by using the “unfreeze” option.

4.1.2. *Unfreeze*

This option removes the boundaries of the fixed selected area and allows the user to change the selected area of interest.

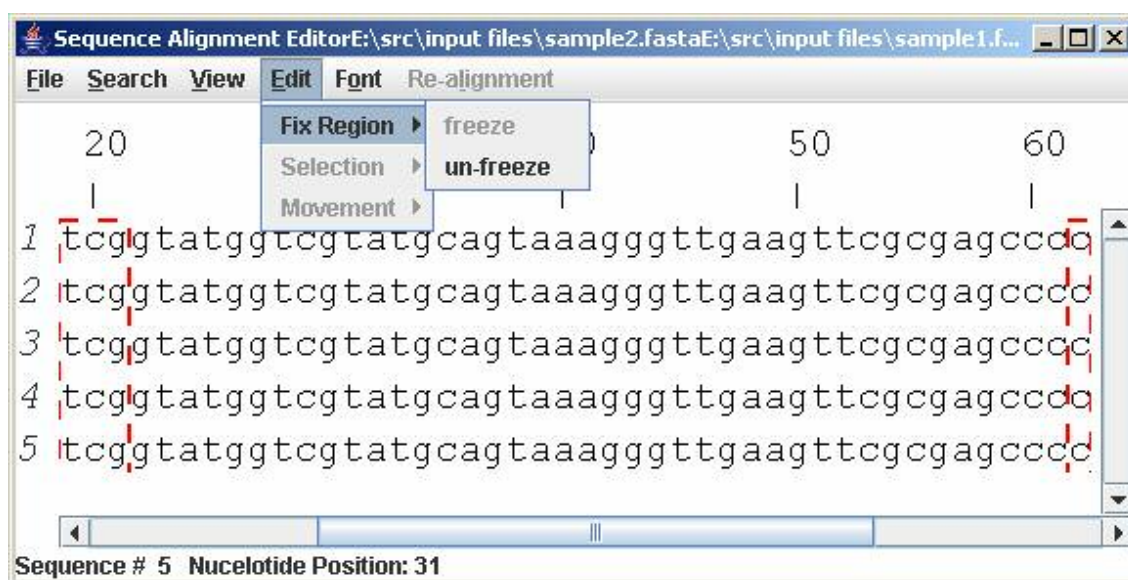


Figure A7 Fix Region menu

4.2. Selection

This menu provides features to handle individual sequences during alignment editing.

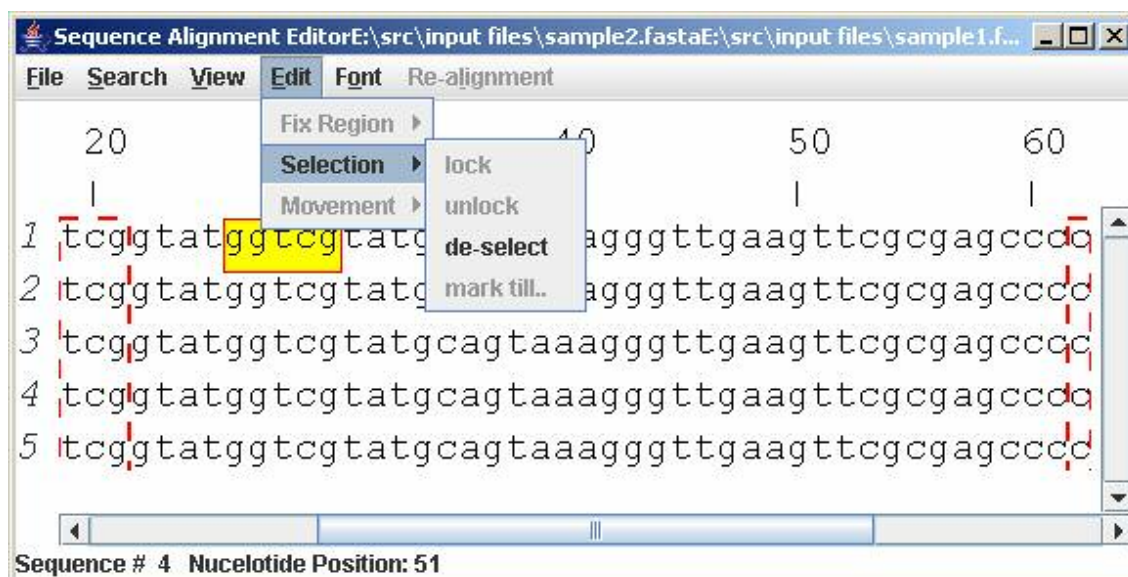


Figure A8 Selection menu

Once the alignment area targeted for editing is frozen, a range of bases called segments from different sequences constituting the alignment, suspected to be aligned with one another should be selected for realigning.

The selection can be done using with a mouse or using the “mark till.” option under the Selection menu.

Such selected range of bases also known as segments appear yellow with a red border. Each sequence can have only one such marked region or segment. The sequence of such a segment marked first, makes the reference sequence. All other sequences will be aligned with respect to the reference sequence.

The Selection menu provides following features to handle marked segments:

4.2.1. *deselect*

This option allows the user to deselect the marked segment and select a different segment for realignment.

4.2.2. *lock*

Sequences marked for realignment can be realigned ONLY IF LOCKED. The lock option locks the sequence. Once a sequence is locked, the marked area cannot be changed unless the sequence is UNLOCKED.

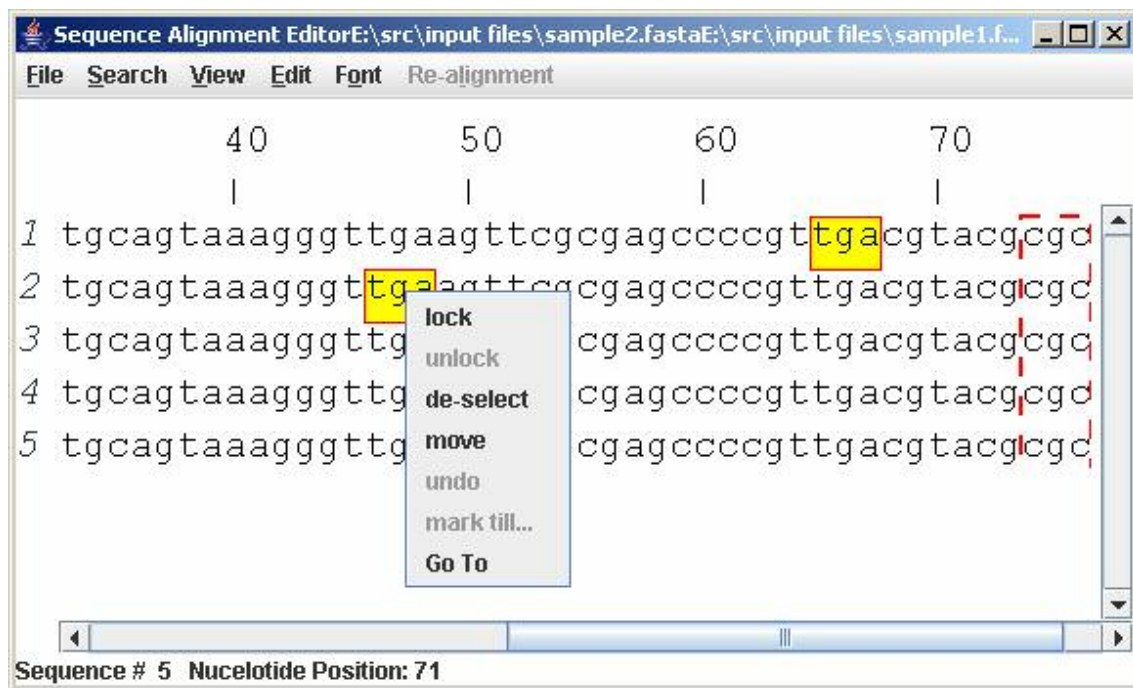


Figure A9 lock option in the pop up menu

4.2.3. *unlock*

This allows the user to change the selected area of a locked sequence.

4.2.4. *mark till*

This option allows user to select a range of bases either to the left or right of the current position for realignment.

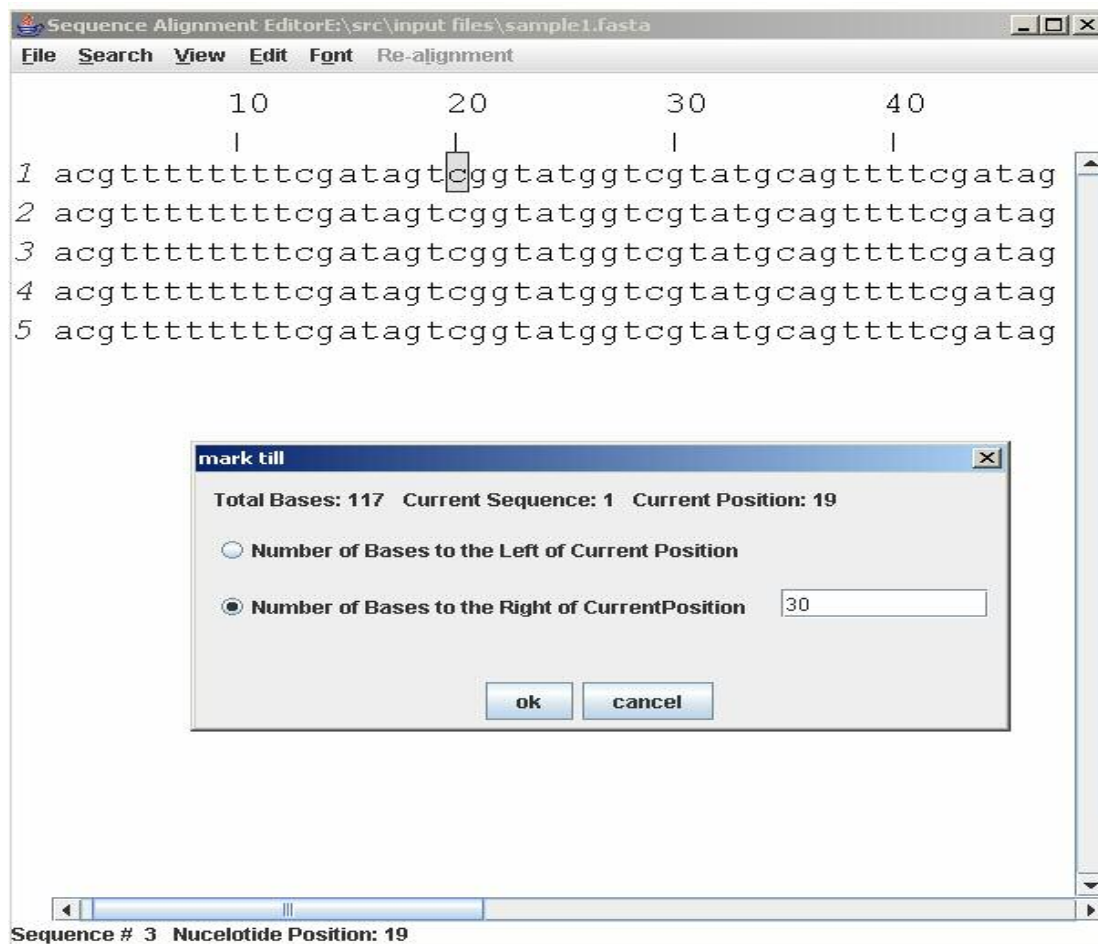


Figure A10 mark till option

4.3. Movement

This menu allows user to bring the marked segments under conformation before realigning them.

Options under Movement:

4.3.1. *Move*

This option brings the marked segment of a sequence (NOT THE REFERENCE SEQUENCE) under the marked segment of the reference

sequence and puts them in conformation. This option works one sequence a time. The same effect is achieved by dragging and dropping the segment under the reference sequence segment using mouse.

4.3.2. *Move all*

“Move All” option moves all the marked segments under the reference sequence segment and lines all the segments under one column in one go.

4.3.3. *Undo*

This option removes a marked sequence from the conformation.

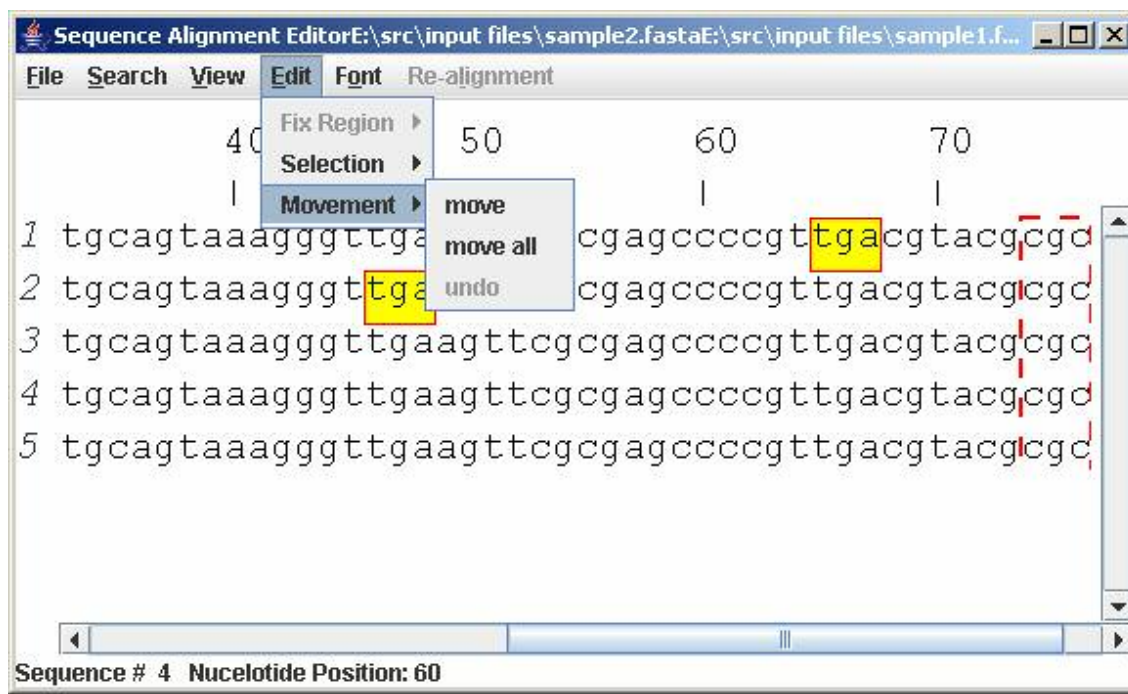


Figure A11 Movement menu

All options under Selection menu except the “Freeze”, ”Unfreeze” and “Move all” options are also provided in a pop up menu that gets displayed when user right clicks on the alignment area.

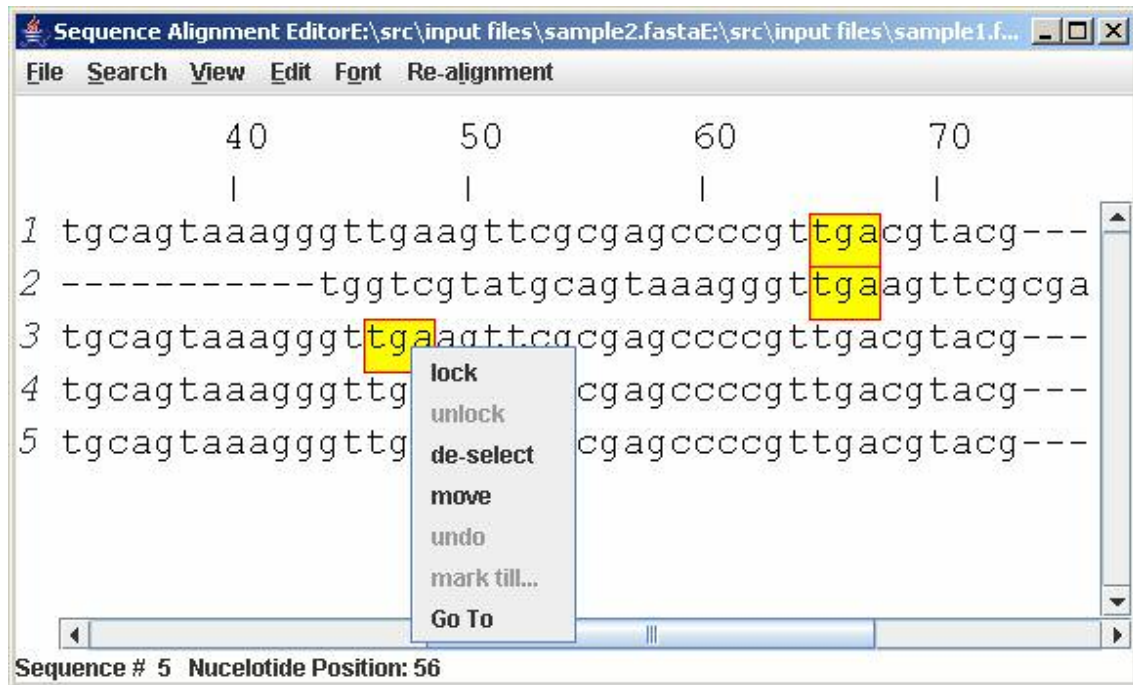


Figure A12 pop up menu

5. Font Menu

This menu gives various font sizes to resize the alignment. Keyboard shortcut for this menu is ALT + F.

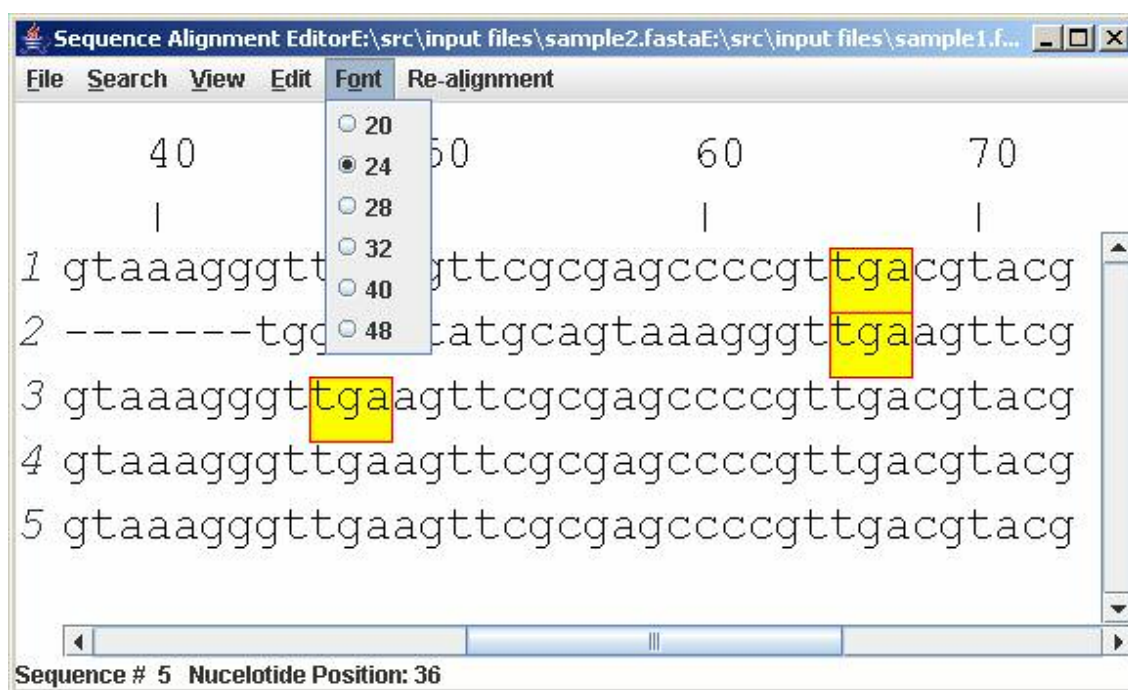


Figure A13 Font menu

6. Re-alignment menu

This menu allows user to realign the sequences after bringing different marked segments under conformation, using the “CLUSTAL W” tool for multiple sequence alignment or cancel out the modifications done in the alignment (by bringing marked segments under conformation). Keyboard shortcut for this menu is ALT + R.

Options available:

6.1. ClustalW

This option feeds the modified alignment to the “Clustal W” tool integrated with the editor, and displays the realigned sequences in the editor.

6.2. Revert

This option revokes all modifications done to the alignment.

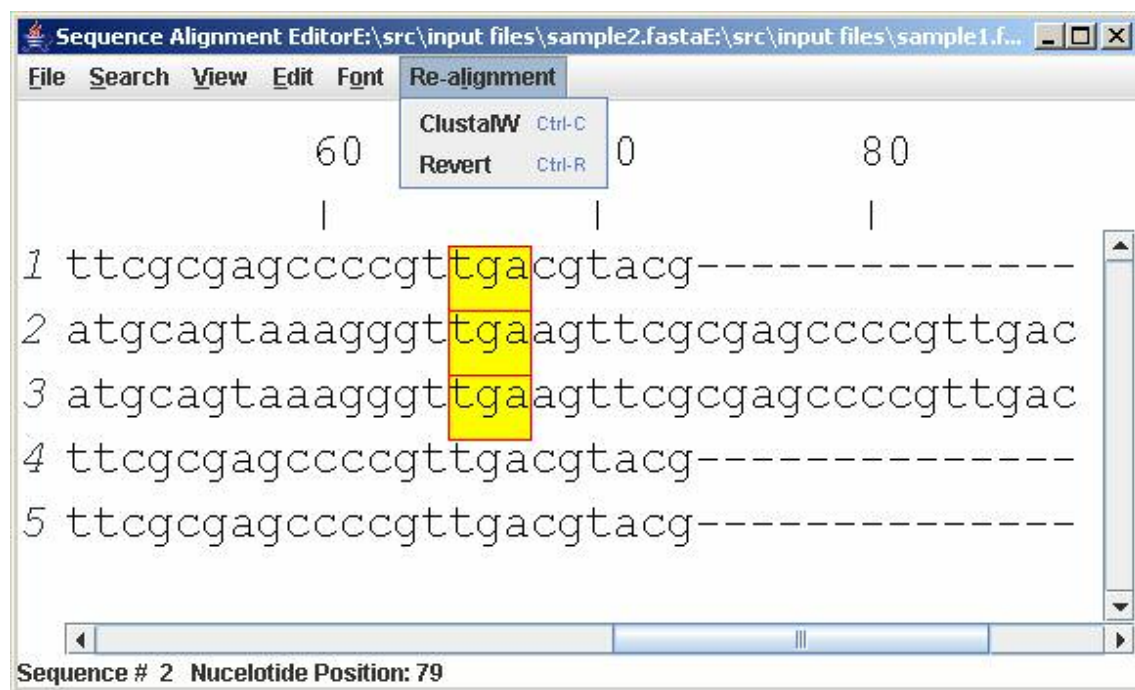


Figure A14 Realignment menu

APPENDIX B

USING *SEQUENCE ALIGNMENT EDITOR*

Sequence Alignment Editor is available for Windows and Linux environments. To use the application, JRE version 1.5 or higher is needed. The application is currently available at http://bioinformatics.uta.edu/alignment_editor/. Instructions to use the tool are given in the website.

REFERENCES

1. A.J. Gibbs, and G.A.McIntyre. The diagram, a method for comparing sequences. Its use with amino acid and nucleotide sequences. *Eur J Biochem.* Sep;16(1):1-11,1970.
2. B. Hamann, and I. Dubchak (2004). Phylo-VISTA: interactive visualization of multiple DNA sequence alignments. *Bioinformatics* 20, 636–643.
3. Bioinformatics: Sequence and Genome Analysis, 2nd ed. David W. Mount. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press, 2004.
4. Bray N and Pachter L, MAVID: Constrained ancestral alignment of multiple sequences, *Genome Research*, 14:693-699 (2004).
5. Carver, T. J., K. M. Rutherford, M. Berriman, M.-A. Rajandream, B. G. Barrell, and J. Parkhill (2005). ACT: the Artemis comparison tool. *Bioinformatics* 21, 3422–3423.
6. Clamp, M., Cuff, J., Searle, S. M. and Barton, G. J. (2004), "The Jalview Java Alignment Editor," *Bioinformatics*, 20, 426-7.
7. Cuff J.A and Barton G.J (1999) Application of enhanced multiple sequence alignment profiles to improve protein secondary structure prediction, *Proteins* 40:502-511.
8. D.F. Feng and R.F. Dolittle. Progressive alignment of amino acid sequences and construction of phylogenetic trees from them. *Methods Enzymol.*266:368-382,1996.
9. D.J. Lipman and W.R. Pearson Rapid and sensitive protein similarity searches. *Science.*1985 Mar 22;227(4693):1435-41.

10. Darling, A. C., B. Mau, F. R. Blattner, and N. T. Perna (2004). Mauve: Multiple alignment of con-served genomic sequence with rearrangements. *Genome Res.* 14, 1394–1403.
11. Edgar, Robert C. (2004), MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research* 32(5), 1792-97.
12. Elnitski, L., C. Riemer, H. Petrykowska, L. Florea, S. Schwartz, W. Miller, and R. Hardison (2002).PipTools: A computational toolkit to annotate and analyze pairwise comparisons of genomic sequences. *Genomics* 80, 681–690.
13. G.J.Barton and M.J. Sternberg . A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons. *J Mol. Biol.*,198,327-337,1987.
14. Galtier, N., Gouy, M. and Gautier, C. (1996) SeaView and Phylo_win, two graphic tools for sequence alignment and molecular phylogeny. *Comput. Applic. Biosci.*, 12, 543-548.
15. Genes VIII- Benjamin Lewin ,Prentice-Hall, Inc., 2004.
16. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A.* 1992 Nov 15;89(22):10915-9.
17. Julie D.Thompson, Desmond G.Higgins and Toby J.Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 1994, Vol. 22, No. 22 46.
18. Katoh,K., Misawa,K., Kuma,K., and Miyata,T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acid Res.*, 30:3059-3066.

18. Mayor C., Brudno M., Schwartz J. R., Poliakov A., Rubin E. M., Frazer K. A., Pachter L. S. and Dubchak I. (2000) VISTA: Visualizing Global DNA Sequence Alignments of Arbitrary Length. *Bioinformatics*, 16:1046.
19. Michael Brudno, Chuong B. Do, Gregory M. Cooper, Michael F. Kim, Eugene Davydov, NISC Comparative Sequencing Program, Eric D. Green, Arend Sidow, and Serafim Batzoglou, LAGAN and Multi-LAGAN: Efficient Tools for Large-Scale Multiple Alignment of Genomic DNA, *Genome Research* 2003 Apr, 13(4): 721 – 31.
20. Michael Brudno, Sanket Malde, Alexander Poliakov, Chuong Do, Olivier Courone, Inna Dubchak, and Serafim Batzoglou .Glocal alignment: finding rearrangements during alignment. Special Issue on the Proceedings of the ISMB 2003, *Bioinformatics* 19: 54i-62i, 2003.
21. Needleman, S.B. and Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, Volume 48, Issue 3, 28 March 1970, Pages 443-453.
22. Parry-Smith, D.J., Payne, A.W.R, Michie, A.D. and Attwood, T.K. (1997) "CINEMA - A novel Colour INteractive Editor for Multiple Alignments." *GENECOMBIS*.
23. Robert C Edgar and Serafim Batzoglou. Multiple sequence alignment. *Current Opinion in Structural Biology*, Volume 16, Issue 3, June 2006, Pages 368-373.
24. Schneider, T., G. Stormo, L. Gold, and A. Ehrenfeucht (1986). Information content of binding sites on nucleotide sequences. *J. Mol. Biol.* 188, 415–431
25. Shannon, C.E. (1948), "A Mathematical Theory of Communication", *Bell System Technical Journal*, 27, pp. 379–423 & 623–656, July & October, 1948.

26. Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, Volume 215, Issue 3, 5 October 1990, Pages 403-410.
27. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, Volume 147, Issue 1, 25 March 1981, Pages 195-19.
28. Wallace, I. M., O. OSullivan, and D. G. Higgins (2005). Evaluation of iterative alignment algorithms for multiple alignment. *Bioinformatics 21*, 1408–1414.
29. Watson J, Crick F (1953). "Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid". *Nature* **171** (4356): 737–8
30. Wilkinson, M., D.Block, and W. Crosby (2002). Genquire: genome annotation browser/editor. *Bioinformatics 18*, 1398–1399.
31. www.jmol.org

BIOGRAPHICAL INFORMATION

Ramya Raghukumar received her Master of Science degree in Computer Science and Engineering from the University of Texas At Arlington in August 2007. She received her Bachelor's degree in Information Technology from The University of Madras, India. Her primary area of interest and research is Bioinformatics. She has developed a new tool called "Sequence Alignment Editor" to help biologists post process sequence alignments. Her future plan is to pursue a career in bioinformatics.