RENYI LIU
Strategies for Improving Multiple Alignment of Retrotransposon Sequences
(Under the Direction of EILEEN T. KRAEMER)

Multiple sequence alignment plays a crucial role in extracting structural, functional, and evolutionary information from the exponentially growing sequence data from the ongoing genome sequencing. Based on the case study of retrotransposon sequence alignment, this thesis compares three alignment programs, DIALIGN, CLUSTALW, and PRRN, and proposes some strategies to improve alignment quality, such as realigning certain sequences or sequence ranges with different programs or parameters and hand editing. Entropy is used as an alignment quality indicator. This study also presents the design and development of an alignment tool, named AlignAgain, which is built to help biologists to improve alignment quality. AlignAgain is written in Java and allows users to display, edit, realign whole or partial sequences with CLUSTALW or PRRN, and append sequences with profile alignment.

INDEX WORDS: Multiple sequence alignment, Java, CLUSTALW, PRRN, Genomics, Retrotransposon, Viewer, Editor.

STRATEGIES FOR IMPROVING MULTIPLE ALIGNMENT OF RETROTRANSPOSON SEQUENCES

by

RENYI LIU

B.S., Wuhan University, P. R. China, 1991

M.S., University of Science and Technology, P. R. China, 1997

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2001

STRATEGIES FOR IMPROVING MULTIPLE ALIGNMENT OF RETROTRANSPOSON SEQUENCES

by

RENYI LIU

Approved:

Major Professor: Eileen T. Kraemer

Committee: John A. Miller
Walter D. Potter

Electronic Version Approved:

Gordhan L. Patel
Dean of the Graduate School
The University of Georgia
August 2001

ACKNOWLEDGMENTS

I would like to thank my major professor Dr. Eileen T. Kraemer for her patience in guiding me, encouraging me, and bearing with me throughout this study. This project can not be done without her invaluable advice on the design and implementation.

My special thanks go to Dr. John Miller and Dr. Don Potter for being on my committee and for opening up new fields of knowledge to me through their classes.

I am very grateful to Dr. McDonald's lab in Genetics department, especially Mr. Eric Ganko for being an invaluable first user of this alignment tool.

My greatest thanks go to my wife, my son, and my parents for their constant love and support.

TABLE OF CONTENTS

Page

CHAPTER 1

INTRODUCTION

With the explosive accumulation of DNA and protein sequence data, especially with the steady progress of genome projects, multiple sequence alignment has become an essential tool in modern molecular biology. Multiple alignments are used to find characteristic motifs and conserved regions in protein families, to detect functional equivalence, to determine evolutionary relationships, and to predict secondary and tertiary structure of gene products. The development of accurate, reliable multiple sequence alignment programs is thus of major importance.

In the last two decades, a number of multiple sequence alignment algorithms, based on various principles, have been developed and extensively used, and new algorithms are constantly proposed. However, when applying these multiple alignment algorithms to certain biological sequences, which often differ in characteristics such as sequence type, identity, length, and substructure, biologists often find it difficult or time consuming to choose the appropriate algorithm and to interact to refine the resulting alignment.

Based on a case study of multiple alignment of retrotransposon sequences, this thesis explores several ways to improve multiple sequence alignment results, including comparison of existing algorithms, conducting alignment on whole or partial sequences with different algorithms or parameters, appending remotely related or new sequences to an existing profile, and hand editing. A graphical user interface based on the latest

visualization and human computer interaction techniques was built to facilitate biologists in the process of finding the best alignment for a concrete application.

Motivation and Goal of the Study

At the beginning of the twenty-first century, there are twenty-four complete genomes available, including sixteen bacterial, six archaeal, and two eukaryotic genomes. In addition, there are estimated to be eighty-two prokaryotic and twenty-four eukaryotic genome-sequencing efforts under way, including that of human genome [Searls 2000]. Tools that identify, store, compare, and effectively analyze a large and growing number of bio-sequences are found of increasingly crucial importance. Multiple sequence alignment is one of the crucial tools that has been extensively studied and widely used. For example, it is becoming a standard practice to search databases with known sequence(s) for homologous sequences, followed by the multiple alignment of the top scoring hits and construction of phylogenic trees.

The existing multiple sequence alignment algorithms may be very different from one another (see review in Chapter 2). When a set of sequences is aligned with different algorithms, or even with the same algorithm but different parameters, the resulting alignments may be very different. Efforts have been taken to help biologists to visualize the alignment results, to evaluate the quality of the alignments and to automate the alignment process. Several graphical tools have been built, such as CLUSTALX [Thompson *et al*. 1997], JalView [Clamp 1998] and CINEMA [Parry-Smith 1997]. However, such tools often have most of the following shortcomings:

- Each tool typically focuses on only one functionality, either alignment or visualization. Users often have to use a standalone alignment program to

produce the alignment, and then feed the alignment to a separate visualization tool.

- The alignment functionality is usually limited to only one alignment program. It is not convenient for users to employ different alignment programs and compare the alignment results.
- Partial sequence alignment is usually not allowed. Different regions of the sequences often differ in sequence identity. Applying the different sets of parameters to different regions may improve the overall alignment quality.
- Hand editing is usually limited and hard to use. Hand editing is the best way to apply user's expert knowledge of features such as sequence structure to the automatic alignment and improve the alignment quality.
- Input and output sequence formats are usually limited. There are currently more than twenty sequence formats in use, but usually only a few are supported.

The goal of this study is to build a graphical tool that overcomes the above shortcomings and helps biologists improve the alignment result for their alignment applications.

CHAPTER 2

BACKGROUND

Multiple Sequence Alignment

Multiple sequence alignment refers to searching for similarity in three or more sequences. In the resulting alignment, homologous residues within the set of sequences are aligned together in columns. Ideally, a column of aligned residues would occupy similar three-dimensional structural positions and all evolve from a common ancestral residue. Figure 1 shows an example of multiple sequence alignment. A formal definition can be defined as follows:

Assume in a set of k ($k > 3$) sequences, $s_1, \ldots, s_k$, each character is taken from an alphabet Σ, which does not contain special gap character "-". A multiple alignment of this set of sequences is a rectangular array, consisting of characters taken from another alphabet Σ', which is Σ plus the gap character "-", that satisfies the following three conditions:

1. There are exactly k rows.
2. Ignoring the gap character, row number i is exactly the sequence s_i.
3. Each column contains at least one character different from "-".

Bio-sequences from the same ancestor are likely to diverge in the evolutionary processes through insertion, deletion and substitution. However, the regions with important structure and functionality are usually more conserved than other regions. The sequence alignments help biologists to identify conserved motifs, which may share

similar structure and functionality. Aligning new sequences to sequences with well

```
GUX1_TRIRE/481-509          HYGQCGGI---GYSGPTVCASGTTCQVLNPYY
GUN1_TRIRE/427-455          HWGQCGGI---GYSGCKTCTSGTTCQYSNDYY
GUX1_PHACH/484-512          QWGQCGGI---GYTGSTTCASPYTCHVLNPYY
GUN2_TRIRE/25-53            VWGQCGGI---GWSGPTNCAPGSACSTLNPYY
GUX2_TRIRE/30-58            VWGQCGGQ---NWSGPTCCASGSTCVYSNDYY
GUN5_TRIRE/209-237          LYGQCGGA---GWTGPTTCQAPGTCKVQNQWY
GUNF_FUSOX/21-49            IWGQCGGN---GWTGATTCASGLKCEKINDWY
GUX3_AGABI/24-52            VWGQCGGN---GWTGPTTCASGSTCVKQNDFY
GUX1_PENJA/505-533          DWAQCGGN---GWTGPTTCVSPYTCTKQNDWY
GUXC_FUSOX/482-510          QWGQCGGQ---NYSGPTTCKSPFTCKKINDFY
GUX1_HUMGR/493-521          RWQQCGGI---GFTGPTQCEEPYICTKLNDWY
GUX1_NEUCR/484-512          HWAQCGGI---GFSGPTTCPEPYTCAKDHDIY
PSBP_PORPU/26-54            LYEQCGGI---GFDGVTCCSEGLMCMKMGPYY
GUNB_FUSOX/29-57            VWAQCGGQ---NWSGTPCCTSGNKCVKLNDFY
PSBP_PORPU/69-97            PYGQCGGM---NYSGKTMCSPGFKCVELNEFF
GUNK_FUSOX/339-370          AYYQCGGSKSAYPNGNLACATGSKCVKQNEYY
PSBP_PORPU/172-200          RYAQCGGM---GYMGSTMCVGGYKCMAISEGS
PSBP_PORPU/128-156          EYAACGGE---MFMGAKCCKFGLVCYETSGKW

consensus                   ...QCGG.......G...C.....C.......
```

Figure 1 An alignment of eighteen cellulose-binding domain of cellobiohydrolase I (CBD-CBH1) sequences. For each sequence, the SWISS-PROT identifier and the position in the parent protein is given on the left. The bottom line shows the consensus, which we define here as the same amino-acid residue type in fourteen or more sequences.

known functionality in different organisms helps us to predict the functionality of new sequences. From multiple sequence alignments, the taxonomic, phylogenetic or cladistic relations among organisms may also be inferred.

Review of Alignment Algorithms

Computationally, multiple alignment is formulated as a combinatorial optimization problem. Due to the complexity of the problem and its importance in modern biological studies, extensive efforts have been devoted to the development of multiple alignment tools. Although numerous methods based on various principles have been proposed and refined, we are still seeking better ones in terms of accuracy and computational speed. Because of the enormous content of this rapidly expanding field, only a brief review on existing algorithms that are related to this thesis is given in the

following section. Extensive reviews are available for further reference [Chan *et al*. 1992; Barton 1998; Gotoh 1999; Thompson *et al*. 1999b].

Although a mathematically optimal alignment of *N* sequences can be achieved by dynamic programming using an *N* dimensional matrix [e.g. Murata *et al*. 1985; Carrillo and Lipman 1988; Gupta *et al*. 1995], the computational complexity makes it impractical for large alignments. To overcome this problem, various heuristic approaches have been developed to rapidly find sub-optimal alignments. Figure 2 shows the major heuristic algorithms, along with the implemented programs that are most commonly used today.

Based on the region on which an alignment is optimized, the algorithms can be categorized as global or local. Global methods construct an alignment throughout the length of the entire sequence, while local methods attempt to identify an ordered series of

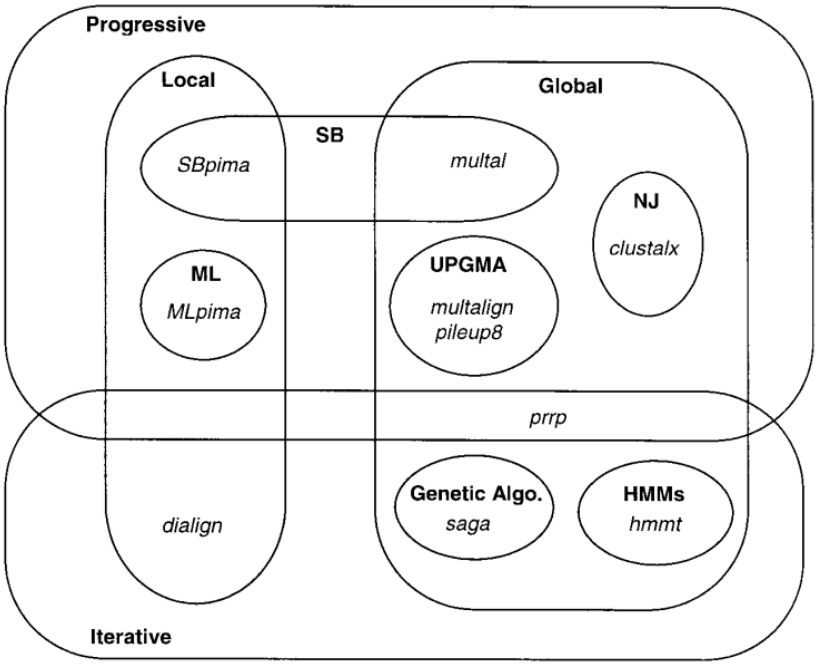

Figure 2 Classification of popular alignment programs and algorithms [Thompson *et al*. 1999b]

motifs and ignore the regions between motifs. A global alignment is stretched over the entire sequence length to include as many matching residues as possible up to and including the sequence ends, and thus may not align identical local regions in order to favor matching more residues along the entire sequence length. Most popular algorithms are global alignment algorithms, for example, DFALIGN [Feng and Doolittle 1987], MULTAL [Taylor 1987, 1988], MSA [Lipman *et al*. 1989], CLUSTALW [Thompson *et al*. 1994], SAGA [Notredame and Higgins 1996], and PRRN [Gotoh 1996]. A local alignment tends to stop at the ends of regions of identity or strong similarity. A much higher priority is given to finding these local regions than to extending the alignment to include more neighboring residues. PIMA [Smith and Smith 1992], MACAW [Schuler *et al*. 1991], and DIALIGN [Morgenstein *et al*. 1996] are local alignment algorithms.

Based on the underlying strategies, the existing algorithms can be categorized as progressive or iterative methods. Progressive approaches begin with an alignment of the most closely related sequences as determined by pairwise analysis and subsequently add the next closest sequence or sequence group to the initial pair. This process is iterated until all sequences have been aligned. CLUSTALW, MULTAL, and MULTALIGN are the well-known examples of progressive algorithms. These algorithms differ in several ways:

- The method to choose the order in which to do the alignment
- Whether the progression involves alignment of sequences to a single growing alignment or if subfamilies are built up on a “guide tree” and then aligned into a whole alignment
- If a “guide tree” is built, the method used to build this tree

Progressive methods do not separate the process of scoring an alignment from the optimization algorithm, neither do they directly optimize any global scoring function of alignment correctness. The major shortcoming of progressive approaches is that once a group of sequences has been aligned, the alignment is “frozen”, thus the errors introduced in the early phase cannot be corrected. The advantage of progressive alignment is that it is fast and efficient, and in many cases the resulting alignments are reasonable, especially when the sequences to be aligned have a high degree of similarity.

Iterative methods overcome the inherent shortcoming in progressive algorithms by using various iterative strategies to refine and improve the initial alignment. The DIALIGN program [Morgenstein *et al*. 1996] constructs multiple alignments based on segment-to-segment comparison rather than residue-to-residue comparison and the segments are incorporated into a multiple alignment with an iterative procedure. The PRRP program [Gotoh 1996, later called PRRN] begins with a progressive, global alignment, and the alignment is refined by iteratively dividing the set of sequences into two groups and realigning using a group-to-group alignment algorithm. SAGA [Notredame and Higgins 1996] uses a genetic algorithm to select an optimal alignment from an evolving population of candidate alignments. It has a unique objective function named COFFEE [Notredame *et al*. 1998]. HMM [Eddy 1995] can be roughly categorized as an iterative method. It is based on hidden Markov models and simulates evolutionary events such as insertion and deletion. It uses the simulated annealing method for optimization. It was reported that iterative methods performed better than progressive methods, especially with remotely related sequences [Gotoh 1996]. Iterative methods are usually computationally more expensive.

CLUSTALW and PRRN

Two representative alignment programs with good performance (see Chapter 3), CLUSTALW and PRRN, are incorporated in the tool built for this study. A brief description of each program is given as follows.

CLUSTALW is a progressive multiple alignment program for DNA or proteins. The basic alignment algorithm consists of three main stages [Thompson *et al.* 1994]:

1) All pairs of sequences are aligned separately in order to calculate a distance matrix giving the divergence of each pair of sequences.

2) A guide tree is calculated from the distance matrix (and can be stored in a file).

3) The sequences are progressively aligned according to the branching order in the guide tree.

CLUSTALW is a stand-alone application that runs either from the command line or through a menu-driven interface. The usability of the tool is significantly improved when combined with the CLUSTALX user interface. The input sequences must be in one file for regular multiple alignment or two files for profile alignment. Although CLUSTALW has its own sequence format, it does support a few other formats, namely, NBRF/PIR, EMBL/SWISSPROT, Pearson (Fasta), GCG/MSF, GCG9/RSF and GDE. All non-alphabetic characters (spaces, digits, punctuation marks) are ignored, except "-", which is used to indicate a gap ("." in GCG/MSF). Over sixty alignment parameters are offered to users to control the alignment.

Pairwise alignment parameters control the speed/sensitivity of the initial alignments, and multiple alignment parameters control the gaps in the final multiple alignments. The sensitivity of the alignment can be improved through sequence weighting, position-specific gap penalties and weight matrix choice. CLUSTALW is well documented and supported, capable of dealing with large numbers of sequences and freely available for Mac, Windows, and various UNIX systems (both source code and executable) from many websites (for example, ftp://ftp.ebi.ac.uk/pub/software/).

The PRRN (Profile-based Randomized iterative Refinement method for alignment of Nucleotide or amino acid sequences) program is an implementation of the randomized iterative strategy for multiple sequence alignment [Gotoh 1996, 1999]. An outline of this algorithm is shown below:

1) Start a preliminary multiple alignment, which can be the output of any multiple alignment program or alignment produced from built-in successive or progressive methods.
2) A phylogenetic tree is constructed from the distance values between members of the current alignment, and a set of weights assigned to all the pairs of sequences is calculated by the three-way algorithm [Gotoh 1995], guided by the phylogenetic tree. A new alignment is obtained with the random iterative refinement method.
3) Repeat step 2) until no change in the total weighted sum-of-pairs score is observed.

This algorithm works most effectively for refining a crude alignment obtained by other more rapid methods, such as progressive alignment. The input sequences can be in

one file or separate files. Only the FASTA sequence format is acceptable for input, but output formats can be others, such as Phylip, GCG, and GDE. PRRN offers only command line and menu interfaces. PRRN program is distributed as ANSI-C source code and is available at ftp.genome.ad.jp/genomenet/saitama-cc free of charge for non-commercial uses. Currently it has only tested by the author on SUN OS 4.1 and Solaris 2.2-5. I found it could not compile successfully on RedHat Linux 7.0.

Quality of Multiple Alignments

The goal of multiple sequence alignment is to seek an optimal alignment for a set of sequences. A quantitative measure is thus desired to assess the alignment results. Since multiple alignment is in principle an optimization problem, alignment algorithms all have explicit or implicit objective functions to optimize. Because sequences in a multiple alignment are related to each other by a phylogenetic tree and some positions in sequences are more conservative than others, an idealized way to score a multiple alignment would therefore be to specify a complete probabilistic model of molecular evolution [Durbin *et al*. 1998]. However, we do not have enough data to build such a complex evolutionary model. Practical scoring systems partially or entirely ignore the phylogenetic tree while doing some sort of position-specific scoring. The following are some commonly used scoring measures.

1. Entropy

The entropy measure is directly related to Shannon entropy in information theory. For an alignment with *n* sequences and *m* columns, the alignment quality for the whole alignment is the sum of the per-column entropies,

$$H = -\sum_{i=1}^{m} \sum_{a} P_a^i \log P_a^i$$

where a is a residue in the alphabet or the gap letter "-", and P_a^i is the frequency of residue a in column i. To capture the fact that the count of a residue in a certain column can be 0 and the alphabet size is different for DNA and protein, the frequency P_a^i is calculated as

$$P_a^i = \frac{c+1}{n+A}$$

where c is the count of residue a in column i, and A is the alphabet size. Gap letters in the same column are treated as different letters to discourage the insertion of gaps in the alignment. The entropy above is the "zero-order" entropy since mononucleotide frequencies are used in the calculation. It can be extended to include higher order entropies to reflect position-specific cost. The alignment that minimizes entropy is assumed to be the "best" alignment for a set of sequences.

2. Sum of Pairs (SP)

Sum of Pairs score is an intuitive and widely used scoring system. It is calculated by summing up the cost of the $n(n-1)/2$ pairs of symbols in each column. The formula is

$$SP = \sum_{i=1}^{m} \sum_{1 \le j < k \le n} w(s_i^j, s_i^k)$$

where n is the number of sequences, m is the number of columns, s_i^j and s_i^k are the letters at row j and k of column i, respectively. Score $w(a, b)$ comes from a substitution matrix such as a PAM or BLOSUM matrix and represents the cost of replacing symbol a by b, or *vice versa*. Gaps are handled by defining $w(-, a)$ and $w(a, -)$ to be the gap cost, and $w(-, -)$ to be zero. The scoring matrices are derived from hand alignment of well-known families and thus contain evolutionary or structural information.

3. Weighted Sum of Pairs (WSP)

Weighted Sum of Pairs score extends SP score by multiplying a weight to each matrix score. It is calculated as

$$WSP = \sum_{i=1}^{m} \sum_{1 \le j < k \le n} \alpha_{j,k} w(s_i^j, s_i^k)$$

where $\alpha_{j,k}$ is a sequence-dependent parameter. The reason to introducing this weight parameter is that in a set of sequences, some sequences are more closely related to one another than to the remaining ones and an appropriately chosen weight parameter can highlight the similarity between closely related sequences and discount over-representation of certain subclasses of sequences. The calculation of the weight parameter can be found in [Perrey *et al*. 1997].

The scoring methods described above can be used to evaluate the "goodness" of any given alignment or compare the quality of two resulting alignments from different alignment runs. For the purpose of comparing the performance of different alignment programs, we can use two other scoring methods if a reference alignment is available. A reference alignment is a "perfect" alignment that has been validated with expert knowledge. The two scoring methods used by Thompson *et al*. [1999] compare a resulting alignment to the reference alignment and measure how "close" they are. For an alignment of *N* sequences consisting of *M* columns, we can designate the ith column in the alignment by A_{i1}, A_{i2}, …, A_{iN}. Define p_{ijk} such that $p_{ijk} = 1$ if residues A_{ij} and A_{jk} are aligned with each other in the reference alignment, and $p_{ijk} = 0$ otherwise. The score S_i for the ith column is defined as

$$S_i = \sum_{j=1, j \ne k}^{N} \sum_{k=1}^{N} p_{ijk}$$

The sum-of-pairs score (SPS) for the alignment is

$$SPS = \sum_{i=1}^{M} S_i \; / \; \sum_{i=1}^{Mr} S_{ri}$$

where Mr is the number of columns in the reference alignment and S_{ri} is the score S_i for the ith column in the reference alignment.

The column score (CS) for the alignment is

$$CS = \sum_{i=1}^{M} C_i \; / \; M$$

where $C_i = 1$ if all the residues in column i are aligned in the reference alignment, and $C_i = 0$ otherwise. The higher the two scores, the closer the test alignment is to the reference alignment.

Comparison of Alignment Programs

Each multiple alignment algorithm has its own advantages and disadvantages. It is thus important for biologists to choose the appropriate program for their application. Several comparisons were conducted to evaluate the relative performance of existing alignment programs. McClure *et al*. [1994] examined nine global and three local multiple protein-sequence alignment methods by applying these programs to four protein families. The criterion is their ability to correctly identify the ordered series of conservative motifs. They found that the performance was affected by the number of sequences in the test sets, the degree of similarity among the sequences, and the number of indels required to produce a multiple alignment. It was concluded that global methods generally performed better than local methods.

Thompson *et al*. [1999] systematically compared the performance of more recently used alignment programs, especially several new iterative algorithms. A benchmark alignment database called BAliBASE was developed specifically for this

purpose [Thompson *et al*. 1999a]. BAliBASE contains 142 validated test alignments of real proteins and can be used to test the performance of alignment programs on sequence sets with different characteristics, such as sequence length, sequence identity, the re-partition of sub-families, and the presence of large terminal extensions and internal insertions. The performance of each program was evaluated by the closeness of resulting alignment to the reference "perfect" alignment, indicated by the sum-of-pairs score (SPS) and the column score (CS). They confirmed the finding of McClure *et al*. [1994] and found that iterative algorithms often offered improved alignment accuracy at the expense of computation time with an exception when a single divergent sequence was introduced into a set of closely related sequences. It was suggested that the employment of more than one program based on different alignment techniques might significantly improve the quality of automatic protein sequence alignment methods.

Alignment Viewers and Editors

An alignment viewer and editor allows users to "interact" with an alignment produced by automatic alignment programs. It provides means for users to display alignment, evaluate alignment quality, identify conservative motifs, and more importantly, apply expert knowledge to refine the alignment. The most popular tools are CLUSTALX and Jalview.

CLUSTALX [Thompson *et al*. 1997] is a user interface for CLUSTALW. It provides means to perform multiple sequence and profile alignment with CLUSTALW, change the order of sequences by cut-and-paste, realign a subset or sub-range of sequences, and analyze alignment quality. In addition, CLUSTALX has versions for all major operating systems. However, it does have some shortcomings: it allows only

automatic alignment with CLUSTALW, it does not allow hand-editing (insert or remove gap letter), and it supports only a few sequence formats.

Jalview [Clamp 1998] is a multiple alignment editor written entirely in java. It displays alignments with several coloring schemes, allows the use of the CLUSTALW program locally or remotely on the EBI server, it provides insertion and deletion of gap letters with a mouse, and it can be run as an Applet. It shares the same problems as CLUSTALX, except gap letter insertion and deletion. Most importantly, it does not allow subset or sub-range sequence realignment.

Alignment of Retrotransposon Sequences

Retrotransposons are the most abundant and widespread class of eukaryotic transposable elements. For example, it is currently estimated that at least 40% of the human genome is comprised of retrotransposons, and this number goes to 50% in maize. Retrotransposons have been acknowledged as major causes of spontaneous mutations, disease, and significant factors in genome evolution. The ongoing genome sequencing of a variety of model experimental organisms and humans is providing an unprecedented opportunity to examine the patterns of molecular variation existing among the entire complement of retrotransposons residing in genomes. Previous analysis of retrotransposons in *Saccharomyces cerevisiae* [Jordan and McDonald 1998] and *Caenorhabditis elegans* [Bowen and McDonald 1999] genomes provided novel insights into the tempo and mode of retrotransposon evolution. Multiple sequence alignment plays a critical role in such analyses. Typically, retrotransposon sequences are retrieved from genome sequence databases by sequence searching program such as BLAST, with known sequence(s). Additional searches are performed with the new sequences until overlapping

hits are retrieved by all sequences. Sequences are then aligned and improved with hand editing, and phylogenetic trees are built for further analysis. CLUSTALX was used to align RT (reverse transcriptase) sequences in the *C. elegans* genome [Bowen and McDonald 1999] and PILEUP was used to align LTR (long terminal repeat) sequences in the *S. cerevisiae* genome [Jordan and McDonald 1998].

CHAPTER 3

COMPARISON OF ALIGNMENT PROGRAMS

To improve the alignment quality of retrotransposon sequences, we must know the relative performance of various alignment algorithms on these sequences. Three programs, CLUSTALW [Thompson *et al*. 1994], PRRN [Gotoh 1996, 1999], and DIALIGN [Morgenstern *et al*. 1996] were selected and their performance compared. The rationale for choosing these programs is:

1. CLUSTALW, PRRN, and DIALIGN are well-known programs, and are representative of progressive, iterative, and local algorithms, respectively.
2. It was reported that these three programs had the best overall performance in their own category based on comparison with the BAliBASE benchmark sequence sets [Thompson *et al*. 1999b].
3. They can be downloaded from the web and installed successfully.

Materials and Methods

Three programs DIALIGN (version 2.0), CLUSTALX (version 1.81), and PRRN (version 3.0) were downloaded from three websites, http://www.gsf.de/biodv/dialign.html, ftp://ftp.ebi.ac.uk/pub/software/, and ftp.genome.ad.jp/genomenet/saitama-cc, respectively. The programs were installed according to their accompanying instructions. Default parameters were used to produce all the alignments with the exception of some sequence input/output options.

Two kinds of test sequences were used. BAliBASE benchmark sequence sets were used to confirm the previous comparison result and to see if entropy is a good

quality indicator of sequence alignments. BAliBASE was downloaded from the website http://www-igbmc/u-strasbg.fr/BioInfo/BAliBASE. Two sets of *C. elegans* retrotransposon sequences, *i.e.* one set of thirty-nine RT (reverse transcriptase) protein sequences and one set of nine LTR (long terminal repeat) DNA sequences, were used to compare the performance of the three programs on the retrotransposon sequence alignment problem.

Since alignments produced by different programs usually have different sequence length, the quality of each alignment was assessed using average column entropy, which is the total entropy value divided by the sequence length. The retrotransposon sequence alignments were also compared to hand-aligned reference alignments with SPS (Sum-of-Pairs Score) and CS (Column Score) scores (entropy, SPS, and CS calculations are described in Chapter 2).

Results

In general, on BAliBASE sequence sets, the average column entropy values of the

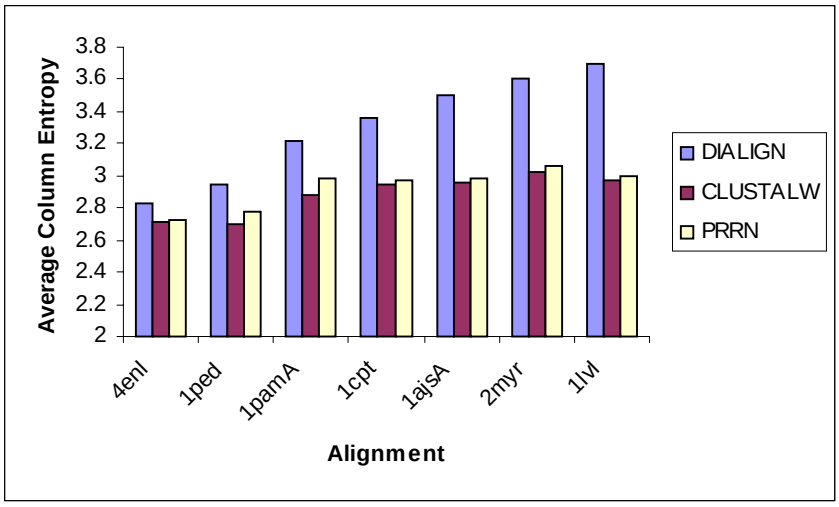

Figure 3 Average column entropy values of the alignments produced by DIALIGN, CLUSTALW, and PRRN with some sequence sets in reference 2 of BAliBASE.

alignments produced by CLUSTALW and PRRN are very close to each other and much lower than that of the alignment produced by DIALIGN. Also, the alignments of DIALIGN are usually longer than the alignments of CLUSTALW and PRRN, and thus have a greater total entropy as well. Figure 3 shows the column entropy values of the alignments produced by the three programs on the same sequence sets in reference 2 of BAliBASE, and Figure 4 shows the sequence length of the same set of alignments. These results are consistent with those of Thompson *et al*. [1999] and suggest that the local alignment algorithm DIALIGN tends to insert more gaps in the alignments. The results also show that entropy value is a sensitive alignment quality indicator with the advantage of not using any score matrix or reference alignment.

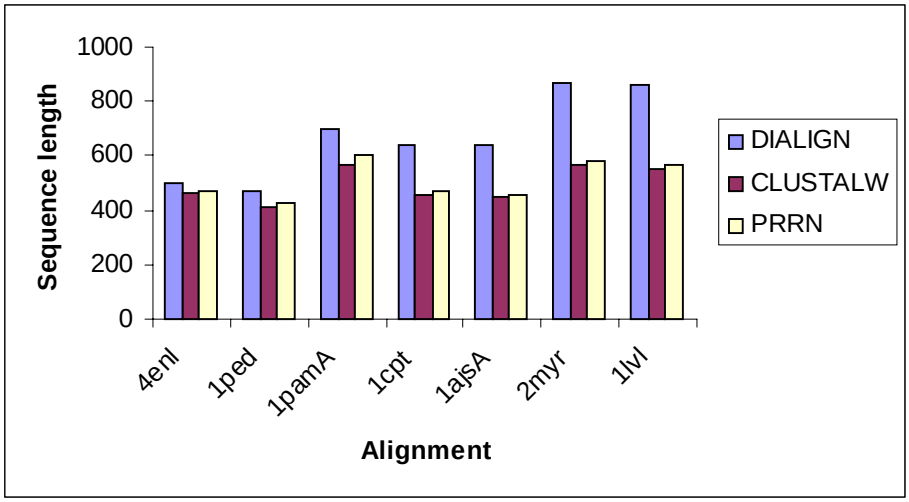

Figure 4 Sequence length of the alignments produced by DIALIGN, CLUSTALW, and PRRN with some sequence sets in reference 2 of BAliBASE.

Alignment results of RT and LTR retrotransposon sequences are very similar to those of BAliBASE sequences (Table 1). Both Sum-of-Pairs Scores and average column entropy values show that the performance of three programs is CLUSTALW > PRRN > DIALIGN, beginning with the best. Since the reference alignments used in the Sum-of-

Pairs Score calculation are hand-edited CLUSTALW alignments and the scores of CLUSTALW alignments are only slightly better than those of PRRN, we see that the performance of CLUSTALW and PRRN are very close to each other.

Table 1 The performance of DIALIGN, CLUSTALW, and PRRN on retrotransposon sequence alignment

	Sum-of-Pairs Score		Average column entropy		Sequence length	
	RT	LTR	RT	LTR	RT	LTR
DIALIGN	0.672	0.682	3.898	2.930	337	688
CLUSTALW	0.845	0.851	3.248	2.731	252	568
PRRN	0.821	0.725	3.323	2.800	260	613

CHAPTER 4

ALIGNMENT TOOL

Functionality

The goal of this study is to find ways to help biologists to improve alignment quality. Through a literature review and interaction with alignment program users, a number of ways are identified, including

- Apply more than one alignment programs on one set of sequences. It might be especially effective to use a progressive program to produce an initial alignment and then use an iterative method to refine the alignment.
- Apply different runs (different programs or the same program with different parameters) to different regions of the sequences. For example, the overall alignment quality may be improved if we produce an alignment with low gap penalties to identify conserved motifs, then realign the regions between motifs with high gap penalties to remove extra gaps introduced in the first step.
- Hand editing. Sequences are derived from the long history of molecular evolution. A reliable alignment must conform to this evolutionary process. Expert knowledge about the conserved residues, secondary and tertiary structure, expected insertion and deletion patterns, and phylogenetic relationships between sequences help biologists to identify and correct errors introduced by automatic alignment methods.

- Profile alignment. Manually remove the outlier sequence(s) from an alignment, realign the remaining sequences, possibly with different parameters, then align the removed sequence(s) back to the profile. The overall alignment quality may be improved. Profile alignment is also desirable to align newly found homologous sequence(s) to an existing alignment, especially in the current situation in which new sequences are constantly produced by ongoing genome projects.

To facilitate users to improve multiple alignments through these strategies, our tool, named AlignAgain, also provide other functionalities:

- It supports fifteen major sequence formats for input and output. The input format is automatically recognized, and the alignments can be written to a file in any of these formats.
- It supports three popular coloring schemes, Zappo, Taylor, and PID, to display the alignments and to help users identify conserved motifs.
- Entropy values are calculated to evaluate the alignment quality.
- Up to five copies of alignments can be kept in memory to help users to compare different alignment.
- Sequence alignment can be performed by alignment programs residing on a remote machine.

Architecture and Implementation

This alignment tool, AlignAgain, was implemented in Java (version 1.3), since Java is an object oriented programming language that provides portability, powerful

graphical user interface and text displaying/editing packages, and a convenient distributed computation mechanism.

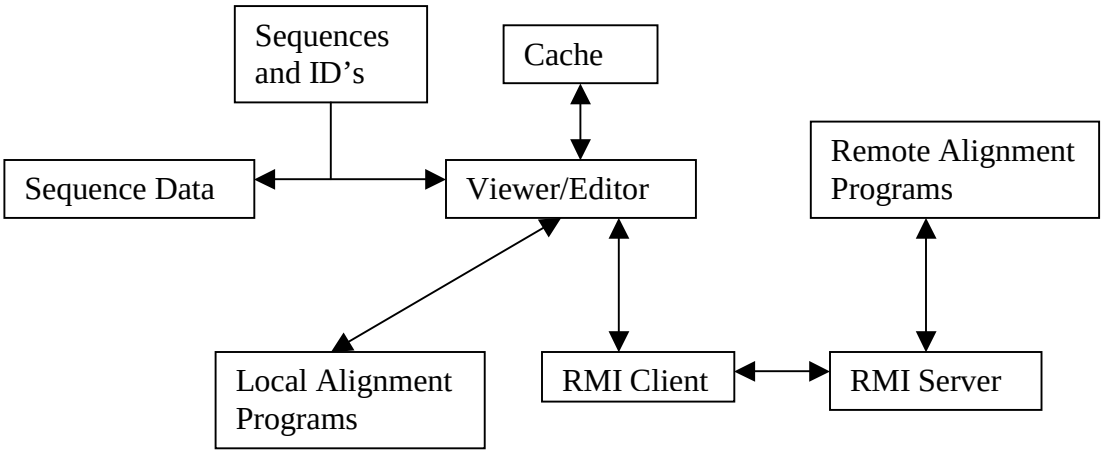

Figure 5 Architecture of AlignAgain

The components (Figure 5) and operating mechanism of AlignAgain system are:

- The original sequences are loaded from local files. Sequence data is cached in memory, but only sequence ID's and residues are visualized.
- To perform a new alignment, the user chooses the alignment program, location, parameters, and sequences to be aligned. The alignment command and the sequences are written to local files or sent to a remote server through RMI (remote method invocation).
- Alignment is performed by invoking the alignment programs, either locally or remotely. The alignment results are sent back and inserted at the appropriate position in the displayed alignment.
- Up to five alignments can be kept in memory to enable undo/redo, of updates to the current alignment.

- The final alignment is written into local files. The updated alignment is incorporated with possible other sequence information cached in memory so that no information is lost.

The Java Swing package defines a complete set of graphical user interface components. The Swing text package has some important features, such as model-view separation, pluggable look-and-feel, scalability, and extensibility [Prinzing 2001] and provides a solid foundation for building text display and editing related applications. The alignment viewer/editor consists of the following components:

- The view. An instance of `JTextPane` that is in charge of displaying sequences.
- The model. An instance of `FASTADocument` that manages the content. The `FASTADocument` class was created to manage sequence specific information. It extends `DefaultStyledDocument` in the Java Swing package.
- The sequence color scheme provider. An instance of newly created `FASTAStyleContext` class, which extends `StyleContext`.

JTextPane enforces automatic linewrapping and has no method provided to disable this feature. It is customized, by overriding its `setSize` and `getScrollableTrackViewPortWidth` methods, so that one sequence is displayed on one single line without linewrapping.

Three sequence color schemes are implemented. Although Zappo (Table 2) and Taylor (Table 3) color schemes were designed for protein sequences, they can be used for

Table 2 Zappo coloring scheme

Residues	Description	Color
ILVAM	Aliphatic/hydrophobic residues	pink
FWY	Aromatic	orange
KRH	Positive	red
DE	Negative	green
STNQ	Hydrophilic	blue
PG	Proline/Glycine	magenta
C	Cysteine	yellow

Table 3 Taylor coloring scheme

Residues	Full name	RGB values
V	Valine	153, 255, 0
I	Isoleucine	102, 255, 0
L	Leucine	51, 255, 0
F	Phenylalanine	0, 255, 102
Y	Tyrosine	0, 255, 204
W	Tryptophan	0, 204, 255
H	Histidine	0, 102, 255
R	Arginine	0, 0, 255
K	Lysine	102, 0, 255
N	Asparagine	204, 0, 255
Q	Glutamine	255, 0, 204
E	Glutamate	255, 0, 102
D	Aspartate	255, 0, 0
S	Serine	255, 51, 0
T	Threonine	255, 102, 0
G	Glycine	255, 153, 0
A	Alanine	204, 255, 0
M	Methionine	0, 255, 0
P	Proline	255, 204, 0
C	Cysteine	255, 255, 0

DNA sequences, due to the fact that they already have a mapping to A, T, C, and G letters. PID (Table 4) is a simple and effective coloring scheme based on sequence consensus and column identity. Residues are colored according to the percentage of the

residues in each column that agree with the consensus sequence. Only the residues that agree with the consensus residue for that column are colored.

Table 4 PID color scheme

Percentage agreement	Color
> 80%	Blue
> 60%	Cyan
> 40%	Light Grey
< 40%	White

Sequence alignment is performed by calling standalone alignment programs CLUSTALW and PRRN. Class `java.lang.Runtime` is used to make program calls. The steps are:

- The sequences are saved as sequence file(s).
- The alignment command is written to a separate file.
- Launch the alignment program by invoking `Runtime.exec()`.
- Read back the alignment result from the output file of the alignment program.

The input parameter of the `exec()` method may be a single string that represents both the program to execute and any arguments to that program, or an array of strings that separate the program from its arguments, or an array of environment variables. The argument to the `exec()` method is also dependent on the operating system. In addition, the standard input and output streams of the sub-process need to be handled properly. Daconta [2000] provided an excellent review on this issue.

The alignment can be conducted by an alignment program residing on a remote machine. However, the alignment parameters and sequences need to be sent to the remote server, which launches an alignment program on that machine. The communication mechanism between the client and the server is RMI.

The readseq package [Gilbert 2001] developed at Indiana University is used to support sequence input and output in various formats.

User View

Figure 5 shows the layout of the graphical user interface (GUI) of AlignAgain. The GUI contains several components. At the top is the menu bar, which contains six dropdown menus, “File”, “Edit”, “Color”, “Alignment”, “Parameters”, “Statistics”, and “Help”. The functions of these menus are:

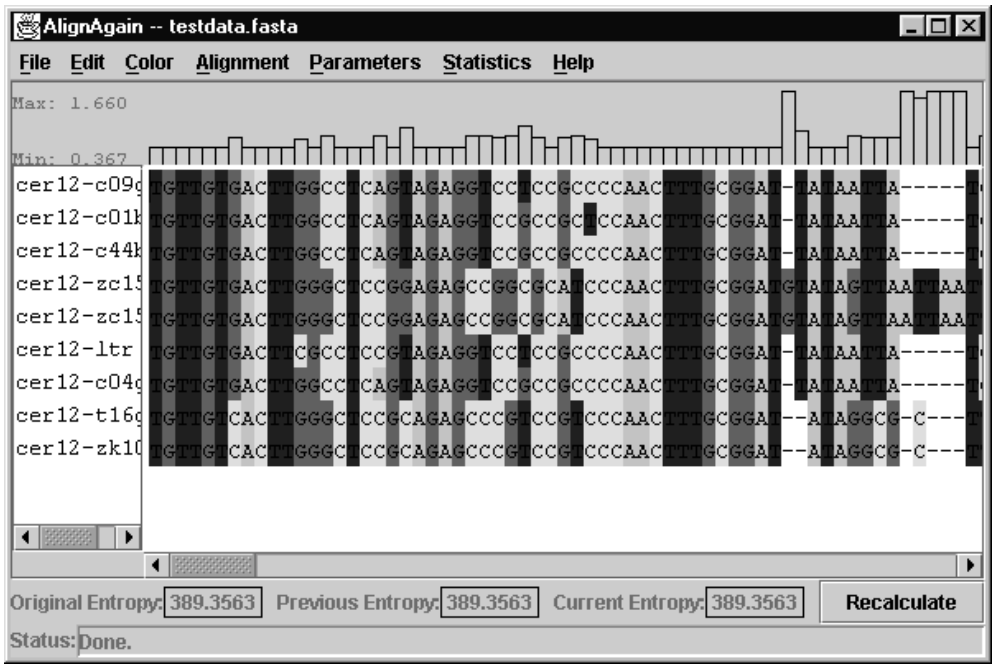

Figure 6 GUI of AlignAgain

- File: sequence input and output.
- Edit: remove the selected sequence(s) and clear sequence selection.

- Color: select from among the three color schemes (Zappo, Taylor, and PID) and choose background / foreground coloring.
- Alignment: choose the alignment program and its location (local or remote), alignment mode (align all sequences, selected sequences, or selected range), and profile alignment mode. Perform alignment. Retrieve previous / next alignment.
- Parameters: check current alignment options, change options, save options to file, read options from file.
- Statistics: provide information about the sequences and alignment quality.
- Help: provide the user guide for this system.

Below the menu bar is the main panel that displays the information about the current alignment. At the top of the main panel is an image panel displaying the small rectangles that represent the relative entropy values for each aligned column. On the left is the list of sequences ID's. On the left-top corner the maximum and minimum column entropy values are displayed. The sequences are displayed in the center.

Two kinds of information are provided at the bottom of the GUI, overall entropy values and status. Original entropy value is the value when the sequence is first loaded. Previous and current entropy values are also displayed. The button "Recalculate" recalculates the entropy value for the current alignment, padding dashes at the end of some sequences if necessary. The status bar gives status information for some time-consuming operations, such as the alignment process.

Details of the user interaction methods are described in the following section.

1. Sequence input and output.

Sequences are loaded from text files. One file must contain all the sequences. Almost all major sequence formats are automatically recognized for input. Alignments can be written to files, in up to fifteen formats: GenBank, EMBL, FASTA, GCG, MSF, Clustal, NBRF, PIR, ACEDB, Phylip, NEXUS, XML, Pretty, DNAStrider, and IG. Users may choose output formats by clicking the menu item "Setup Output Format …" under the "File" menu, which brings up the format panel (Figure 7). Users may choose to save all sequences or selected sequences only with corresponding menu items under the "File" menu. Sequences can be selected by selecting sequence ID's on the left of the main panel. With the "shift" or "ctrl" key pressed, multiple selections can be made.

Figure 7 Sequence output format panel

2. Hand Editing.

Users can edit the current alignment by inserting or deleting the gap letter "-". To delete, the user puts the cursor before the gap letter, then presses the "Delete" key, or

presses “D” with the “ctrl” key held down. Any of the three keys, “Space”, “minus”, and “subtract”, may be used to insert a gap letter. Four arrow keys, “Left”, “Right”, “Up”, and “Down”, are used to navigate through the alignment. The “Home” key brings the cursor to the beginning of the alignment, and the “End” key brings the cursor to the end. During gap letter insertions and deletions, whenever all the sequences have the same length, the entropy value is recalculated and the column entropy panel is updated. When the sequences do not have the same length, users can press the “Recalculate” button at the bottom right corner of the GUI to pad gap letters at the end of the sequences and update the entropy calculation. The user can also remove selected sequences with the corresponding menu item under the “Edit” menu.

3. Coloring schemes.

The user may change the color scheme of the current alignment at any time. Menu items under the “Color” menu provide the ability to switch background and foreground coloring and among the three color schemes, Zappo, Taylor, and PID.

4. Regular alignment with local programs.

Users may perform two kinds of alignments: regular or profile alignment. Regular alignment has only one input sequence file. Profile alignment has two input sequence files. Sequences from profile 2 are appended to profile 1.

Regular alignment with local programs consists of the following steps:

- Step 1: Choose the alignment program, CLUSTALW or PRRN, and set the program location to “Local”. This can be done by clicking the corresponding radio buttons under the Alignment→Program and Alignment→Program Location menu items.

- Step 2: Choose the alignment parameters. The user may modify the current or default alignment parameters through the CLUSTALW or PRRN parameter panel (Figure 8 and Figure 9). After the parameters are set, the user may click the "Apply" button to apply the parameters to the current alignment session, or click the "Save" button to save the parameter set to a file and read it back later. When the "Apply" or "Save" button is clicked, parameter values are checked for type and range, if an error is caught, the user is notified to change it.

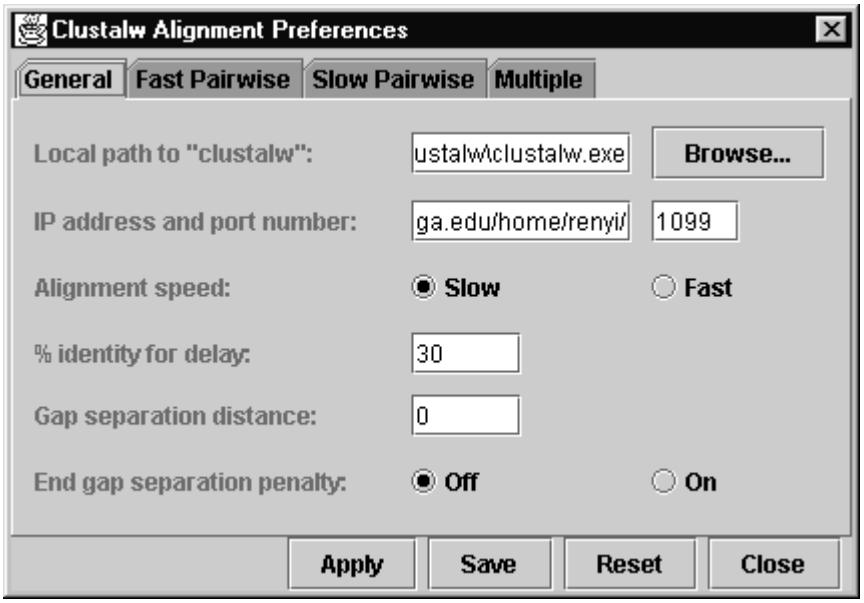

Figure 8 CLUSTALW alignment parameter panel

When AlignAgain is run the first time on one machine, the user needs to specify the path to the alignment program through the parameter panel.

- Step 3: Specify sequences to be aligned. From the menu item Alignment→Regular Alignment Mode, the user may choose to align whole sequences, selected sequences, or a selected range. If the user chooses to align selected sequences, the selection can be made by clicking sequence IDs on the left sequence ID list. If the user chooses to align a selected range, the user may select

a certain range of sequences by mouse dragging. The selected range will be highlighted in light blue. It can be unselected by clicking anywhere on the sequence panel.

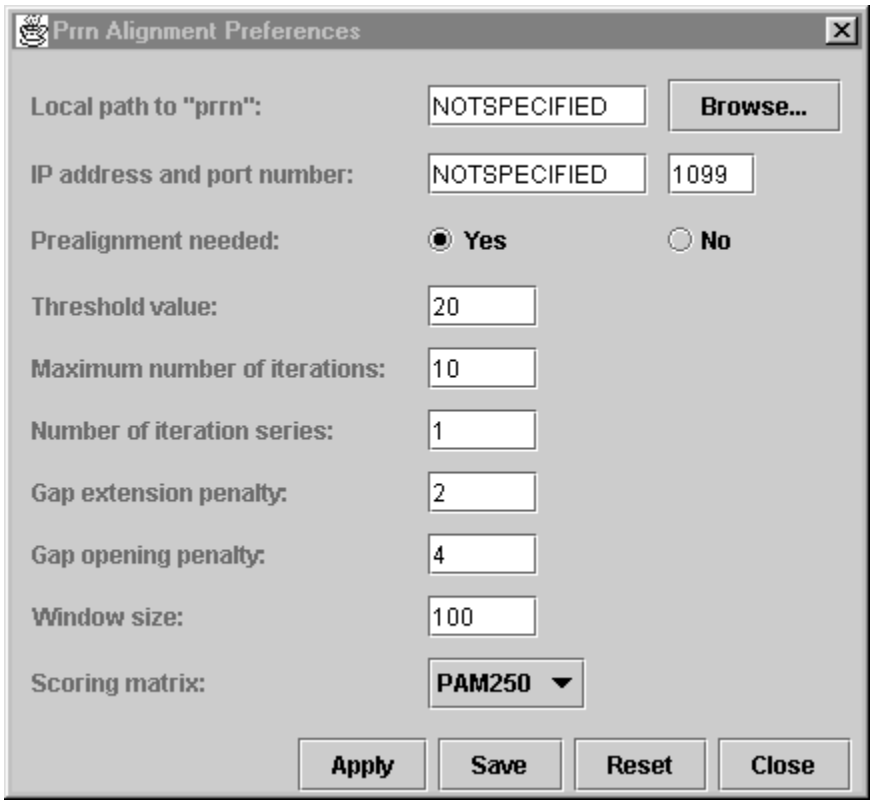

Figure 9 PRRN alignment parameter panel

- Step 4: Click the menu item “Do Alignment” under the “Alignment” menu to perform the alignment. The alignment result is automatically displayed in the sequence panel. If the aligned sequences are whole sequences or a selected range, they are inserted back to the appropriate location. If selected sequences are aligned, the alignment result is treated as a new alignment separate from the previous one and displayed in the same sequence panel. The alignment result is highlighted in green when inserted back into the display. The highlight can be removed by clicking anywhere in the sequence panel.

5. Profile alignment.

To do a profile alignment, the user needs to load another set of sequences as profile2 by clicking the menu item File→Load Sequences to Profile2. This set of sequences is displayed in a separate panel as shown in Figure 10. After choosing the alignment program and parameters, the user may align all sequences or selected sequences from profile2 to the current profile in the main panel by clicking the corresponding menu item under “Alignment”.

6. Sequence alignment with remote programs.

The procedure to align sequences with remote programs is similar to that with

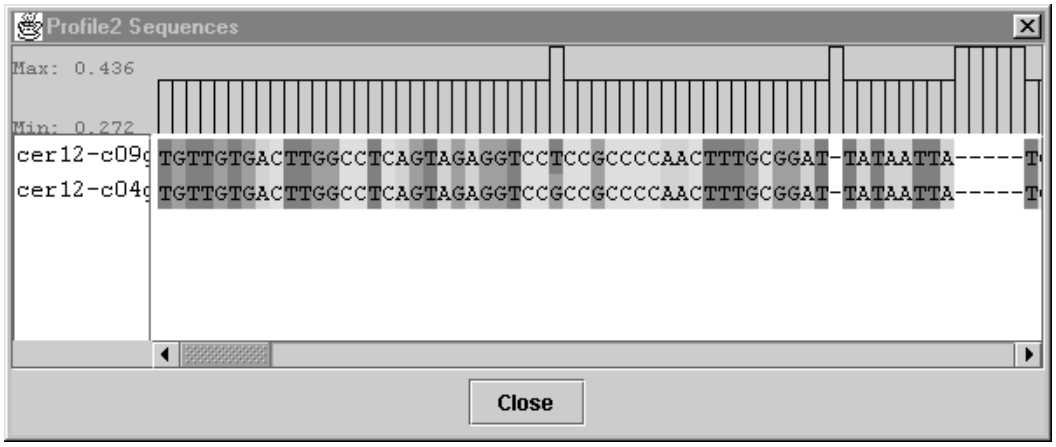

Figure 10 Profile2 panel that displays the second set of sequences for profile alignment

local programs, except that user must set the address of the remote machine through the parameter panel and have the remote server running first.

7. Sequence statistics.

Two kinds of information about sequences in the current alignment are provided. One is about the sequence composition: the A, T, C, G percentages of the sequences (useful for DNA only) are displayed when the menu item Statistics→Sequences is

clicked. The other is about the quality of the alignment: the overall entropy value is displayed on the bottom of the sequence panel and per column entropy values are displayed as rectangles above the sequence panel. Maximum and minimum column entropy values are displayed at the left top corner. Entropy values can be displayed on a separate panel when the menu item Statistics→Entropy is clicked.

CHAPTER 5

EVALUATION OF THE ALIGNMENT TOOL

The alignment tool is designed to provide biologists an environment in which to conduct sequence alignment, view the alignment results, and improve the alignment in an interactive way. It is thus crucial to have biologists involved in the whole development process. From the feedback of the biologists, we have a better idea on how the alignments are conducted, the features that should be built into an alignment tool, and the user interaction methods that are needed.

The first user of this alignment tool is a graduate student in the Genetics Department who has been doing research on retrotransposon evolution. Reliable retrotransposon sequence alignment plays a crucial role in his study. User evaluation was conducted periodically. During each evaluation, the user was told about what features and interaction methods were updated in the system, and then the user was asked to try the system. The user was then asked about how he liked the features and interaction methods, and how the tool may be improved. New ideas came out from almost every evaluation.

When the tool was first built, it had only the basic functions, such as displaying the alignment, performing alignment on whole sequences and a selected range of sequences, and calculating entropy values. Then the user suggested that one of the features he wanted most was hand editing, since currently he had to do alignment with one tool and edit the alignment with another. In the late evaluations, several other features were suggested by the user:

- Provide sequence statistics. For the DNA sequences he is working on, ATCG percentages in each sequence help him to identify sequences with unusual characteristics and poorly aligned sequences.
- Align selected sequences and profile alignment. The user wants to improve alignment by aligning closely related sequences first, then aligning other sequences to the existing profile. Furthermore, the user will work on alignment for the retrotransposon sequences from genomes under sequencing. It is necessary to align newly retrieved sequences to an existing alignment while keeping a hand-edited profile.
- Alignment with programs on a remote machine. This feature was motivated by the fact that PRRN program could only be successfully installed on a Solaris system. Without remote program execution, AlignAgain would have to be running on the same system to use the PRRN program for alignment.
- Padding gap letters after hand editing. During the evaluation, right after the hand editing function was added to the system, the user tried some editing and found he had to manually add some gap letters at the end of some sequences to make every sequence have the same length. He said that this process could be time consuming if the number of sequences was large and suggested the addition of a padding function to make this process automatic.

A formal user evaluation is currently in progress. The goal of the formal evaluation is to evaluate this tool in terms of the goodness of the final alignment and the time it takes to reach the final alignment, as well as, the convenience of the interaction methods.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Multiple sequence alignment plays a crucial role in biological studies. Although numerous multiple alignment programs based on various principles are available, biologists are still seeking ways to improve alignments in terms of accuracy and computation speed. Using retrotransposon sequence alignment as a case study, we investigated the process of alignment. Three representative alignment programs, CLUSTALW, PRRN, and DIALIGN were compared using real retrotransposon sequences and sequences from the benchmark database BAliBASE as test data and entropy as quality measure. In general, CLUSTALW and PRRN performed better than DIALIGN. By interacting with biologists, several ways to improve an alignment were identified, including applying different programs and different parameter sets to the alignment, realigning certain regions in the sequences, removing badly aligned sequence(s) then appending by profile alignment, and hand editing. An alignment tool, AlignAgain, was built to help biologists to improve multiple sequence alignments. This tool was written entirely in Java, and supports many sequence formats, several color schemes, and remote program executions.

To make this alignment tool more powerful and more user-friendly, the following features may be implemented in future work:

- Save alignments to PostScript (PS) or Portable Data Format (PDF) format files. Currently, alignments can only be saved as text files in various sequence formats

and the color information is thus lost. To identify the conservative motifs, those alignments must be loaded into a viewer or editor. PS or PDF format files are also convenient for data exchanging and publishing.

- Print and print preview. Currently users have to save alignments as text file, then print with other facilities.
- Enable calls to other alignment programs. Other programs, such as DIALIGN, may have better performance on certain concrete sequence sets and may be applied to certain regions of sequences to improve overall quality.
- Hand editing undo/redo.
- Database search and alignment updating. This would allow users to use the current alignment to search a local or remote sequence database and align new sequences.

REFERENCES

Barton, G.J. 1998. Protein sequence alignment techniques. *Acto Cryst* D54, 1139-1146.

Bowen, N.J. and J.F. McDonald. 1999. Genomic analysis of Caenorhabditis elegans reveals ancient families of retroviral-like elements. *Genome Research,* 9:924-935.

Carrillo, H. and D. Lipman. 1988. The multiple sequence alignment problem in biology. *SIAM J. Appl. Math*. 48:1073-1082.

Chan, S.C., A.K.C. Wong, and D.K.Y. Chui. 1992. A survey of multiple sequence comparison methods. *Bull. Math. Biol*. 54:563-598.

Clamp, M. 1998. Jalview documentation.

http://circinus.ebi.ac.uk:6543/jalview/contents.html

Dacunta, M. C. 2000. When Runtime.exec() won't.

http://www.javaworld.com/javaworld/jw-12-2000/jw-1229-traps.html.

Durbin, R., S. Eddy, A. Krogh, G. Mitchison. 1998. Multiple sequence alignment methods. In: *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. pp.134-159. Cambridge University Press.

Eddy, S. R. 1995. Multiple alignment using hidden markov model. *Proc. Third Int. Conf. Intelligent Systems for Molecular Biology*, C. Rawlings et al., eds. AAAI Press, Menlo Park. pp. 114-120.

Feng, D.F. and R.F. Doolittle. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J.Mol. Evol*. 21:112-125.

Fuellen, G. 2001. VSNS BioComputing Division Multiple Alignment Resource Page. http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/welcome.html

Gilbert, D. 2001. readseq package. http://iubio.bio.indiana.edu/soft/molbio/readseq/java/

Gotoh, O. 1995. A weighting system and algorithm for aligning many phylogenetically related sequences. CABIOS, 11:543-551.

Gotoh, O. 1996. Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J. Mol. Biol*. 264: 823-838.

Gotoh, O. 1999. Multiple sequence alignment: algorithms and applications. *Adv. Biophys*. 36: 159-206.

Gupta, S. K., J. D. Kececioglu, and A. A. Schaffer. 1995. Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *J.Comp. Biol*. 2(3):459-472.

Jordan, I.K. and J.F. McDonald. 1998. Evidence for the role of recombination in the regulatory evolution of *Saccharomyces cerevisiae* Ty elements. J. *Mol. Evol*. 47:14-20.

Lipman, D. J., S. F. Altschul, and J. D. Kececioglu. 1989. A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA*, 86: 4412-4415.

McClure, M. A., T. K. Vasi, and M. F. Walter. 1994. Comparative analysis of multiple protein-sequence alignment methods. *Mol. Biol. Evol*., 11(4):571-592.

Morgenstern, B., K. Frech, A. Dress, and T. Werner. 1996. Multiple DNA and protein sequence alignment based on segment-to-segment comparison. *Proc. Natl. Acad. Sci. USA* .14: 290-294.

Murata, M., J. S. Richardson and J. L. Sussman. 1985. Simultaneous comparison of three protein sequences. *Proc. Natl. Acad. Sci. USA*. 82:3073-3077.

Notredame, C. and D. G. Higgins. 1996. SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Reseach*, 24(8): 1515-1524.

Notredame, C., L. Holm, and D. G. Higgins. 1998. COFFEE: an objective function for multiple sequence alignments. *Bioinformatics*. 14(5): 407-422.

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. *Gene*, 211(2):GC45-56. Webpage: http://www.bioinf.man.ac.uk/dbbrowser/CINEMA2.1/

Perrey, S.W., J. Stoye, V. Moulton, and A.W.M. Dress. 1997. On simultaneous versus iterative multiple sequence alignment. Preprint.

Prinzing T. 2001. Using the Swing text package. http://java.sun.com/products/jfc/tsc/articles/text/overview/

Robinson, M. and Vorobiev. 1999. Swing. http://manning.spindoczine.com/sbe/

Schuler, G. D., Altschul, S. F., and Lipman, D. J. 1991. A workbench for multiple sequence alignment construction and analysis. *Proteins Struct. Funct. Genet*. 9: 180-190.

Searls, D. B. 2000. Bioinformatics tools for whole genomics. *Annu. Rev. Genomics* Hum. *Genet.* 01:251-279.

Smith, R.F. and T.F. Smith. 1992. Pattern-induced multi-sequence alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modeling. *Protein Engng*. 5(1):35-41.

Taylor, W. R. 1987. Multiple sequence alignment by a pairwise algorithm. *CABIOS*, 3:81-87.

Taylor, W. R. 1988. A flexible method to align larger numbers of biological sequences. *J. Mol. Evol*. 28:161-169.

Thompson, J. D., D. G. Higgins, and T. J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673-4680.

Thompson, J.D., T.J. Gibson, F. Plewniak, F. Jeanmougin, and D.G. Higgins. 1997. The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucl. Acids. Res.* 25: 4876-4882

Thompson J.D., F. Plewniak, and O. Poch. 1999a. A benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15: 87-88.

Thompson, J. D., F. Plewniak, and O. Poch. 1999b. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, 27(13):2682-2690.