Using a Tabu Search Approach to Align DNA Sequences

Carin Lightner-Laws, PhD

Consultant, Bonnie Speed Logistics

and

Constance Lightner, PhD

Associate Professor, Fayetteville State University, Department of Management

INTRODUCTION

DNA Sequencing

DNA sequencing can show the evolution of sequences that diverged from common ancestors over time. Although the most important regions of DNA are usually conserved to ensure survival, slight changes or mutations do occur as sequences evolve [37]. These mutations are any combination of insertion, deletion and/or substitution events. Methods such as sequence alignment are used to detect and quantify similarities between different DNA and protein sequences that may have evolved from a common ancestor. Sequence alignment gives insight into the structure and functions of a sequence, shows a common ancestry or homology between sequences, detects mutations in DNA that lead to genetic disease and is the first step in constructing phylogenetic or evolutionary trees.

Sequence alignment (SA) is an optimal way of inserting dashes into sequences in order to minimize (or maximize) a specified scoring function [1, 37]. Generally, genetic sequencing can be classified as a pairwise sequence alignment or a multiple sequence alignment (MSA). MSA is simply an extension of pairwise alignments that align 3 or more sequences. Both MSA and pairwise SA can further be categorized as global or local methods. Whereas global methods

attempt to align entire sequences, local methods only align conserved regions of similarity.

Multiple Sequence Alignment

Multiple sequence alignments are often classified as progressive or iterative. Typically progressive alignments involve three steps. In the first step, each pair of sequences is aligned using a Dynamic Programming approach. Then, the scores from the pairwise alignments in step one are used to construct a tree. Finally, the tree from step two is used to progressively align the sequences and calculate an alignment score. In progressive algorithms, once a gap is introduced in the early stages of an MSA it is always present; thus one major drawback is that an error in an initial subalignment will be propagated throughout the entire MSA [26]. To avoid these problems, iterative techniques are used and initial alignments are constantly modified.

In many MSAs, a distance score is used to construct a tree. A distance score represents the number of changes required to change one sequence into another. In theory, the score reflects the amount of evolutionary time that has elapsed since the sequences diverged from a common ancestor; thus, a larger distance score, indicates greater evolutionary time and more sequence divergence [26]. The simplest way to calculate the distance score is to sum up the number of mismatches in an alignment and divide by the total number of matches and mismatches.

LITERATURE REVIEW

ClustalW is a commonly used multiple sequence alignment program [19, 40, 20]. As with any other heuristic, ClustalW does not guarantee an optimal solution. It progressively aligns sequences and exploits the fact that similar sequences are evolutionarily related. First, ClustalW aligns and scores all possible pairs of sequences to determine their distance score. Then a guide tree is constructed using the edit distances and a neighbor joining algorithm.

Finally, the guide tree is used to progressively align the sequences. Although an optimal solution is not guaranteed, ClustalW usually provides a good starting point for other refinement methods such as Hidden Markov Modeling. Since ClustalW progressively aligns sequences, any regions that were misaligned early in the process cannot be corrected as the program progresses and more information is introduced. Another problem with ClustalW is the choice of alignment parameters. Sequences that are not highly conserved or not very similar are extremely sensitive to the adjustments of the parameters. When aligning divergent sequences, slight parameter adjustments will drastically change the final multiple sequence alignment. In general, it is difficult to justify why one scoring matrix or parameter selection is better than another [37]. As a direct result of the uncertainty involved in selecting the parameters, ClustalW is most useful when sequences are known to be evolutionarily related.

Notredame and Higgins [29] have the best known genetic algorithm, Sequence Alignment by Genetic Algorithm (SAGA), for multiple sequence alignment. Similar to other genetic algorithms (GA), SAGA uses the principles of evolution to find the optimal alignment for multiple sequences. This method generates many different alignments by rearrangements that simulate gap insertion and recombination events to generate higher and higher scores for the MSA [26]. In this GA, the population consists of alignments that were formed from a complex set of twenty two different crossover and mutation operations. To determine the fitness of an alignment, SAGA uses a weighted sum of pairs approach in which each pair of sequences is aligned and scored; then the scores from all the pairwise alignments are summed to produce an alignment score. As with any heuristic approach, SAGA may not generate an optimal MSA. Although it has been shown that SAGA does produce quality alignments, the time complexity involved in the weighted sum of pairs fitness function is a major drawback to

this approach.

Brudno et al. [5] propose a glocal alignment approach that combines features of both local and global alignment methods. This glocal aligner, Shuffle-LAGAN (SLAGAN), is a pairwise alignment algorithm that can be extended to multiple sequence alignment. The distinct differences between global and local alignments are merged in the SLAGAN approach. Whereas global alignments transform one sequence into the other by using a combination of insertions, deletions and substitutions (simple edits), local alignment techniques tend to focus on similarities between conserved regions; this glocal alignment combines these features and creates a map that transforms one sequence into the other while allowing for rearrangement events. SLAGAN includes rearrangement events because DNA is known to mutate by simple edits, rearrangements such as translocations (a subsegment is removed and inserted in a different location but with the same orientation), inversions (a subsegment is removed from the sequence and then reinserted in the same location but with the opposite orientation and duplications (a copy of a subsegment is inserted into the sequence and the original sub-sequence remains unchanged) or any combination of these simple edits. SLAGAN quickly aligns long sequences. In this technique, a penalty is incurred for the set of operations that include insertions, deletions, point mutations, inversions, translocations and duplications. This approach minimizes the sum of these penalties (edit distance). SLAGAN has three distinct stages. The first stage consists of finding local alignments using the CHAOS tool. The second stage picks the maximal scoring subset of the local alignments under certain gap penalties to form a 1-monotonic conservation map. Whereas standard global alignments are non-decreasing in both sequences, the structure of the 1-monotonic conservation map is non-decreasing in one sequence and without restrictions in the second sequence. Relaxing this assumption in the

second sequence, allows the algorithm to detect rearrangements. The inclusion of rearrangement events is one of the features that make this algorithm different than typical global alignment approaches. In the final stage of SLAGAN, the conservation map of local alignments is joined to form maximal consistent subsegments that are aligned using the LAGAN global aligner. One drawback of the SLAGAN algorithm is that it is not symmetric in the sequence order, so it frequently misses duplications in the sequences.

NEW APPROACH TO MULTIPLE SEQUENCE ALIGNMENT

Basic Components of a Tabu Search

Tabu search is a heuristic approach that uses adaptive memory features to align multiple sequences. The adaptive memory feature, a tabu list, helps the search process avoid local optimal solutions and explores the solution space in an effective manner [33]. While the tabu list restricts the search of some neighboring solutions, there are conditions which allow exceptions to the tabu list. As displayed in Figure 1, a tabu procedure starts with an initial solution, generates neighbors and then moves to the best accessible neighboring solution. Accessible solutions are either not on the tabu list or are on the tabu list but satisfy a condition, an aspiration criterion, which allow exceptions to the tabu list [9]. In cases where the tabu search becomes stabilized and is no longer moving toward better solutions, a diversification and intensification procedure is implemented, so that more of the solution space can be explored. These steps are repeated, until some termination criterion is met.

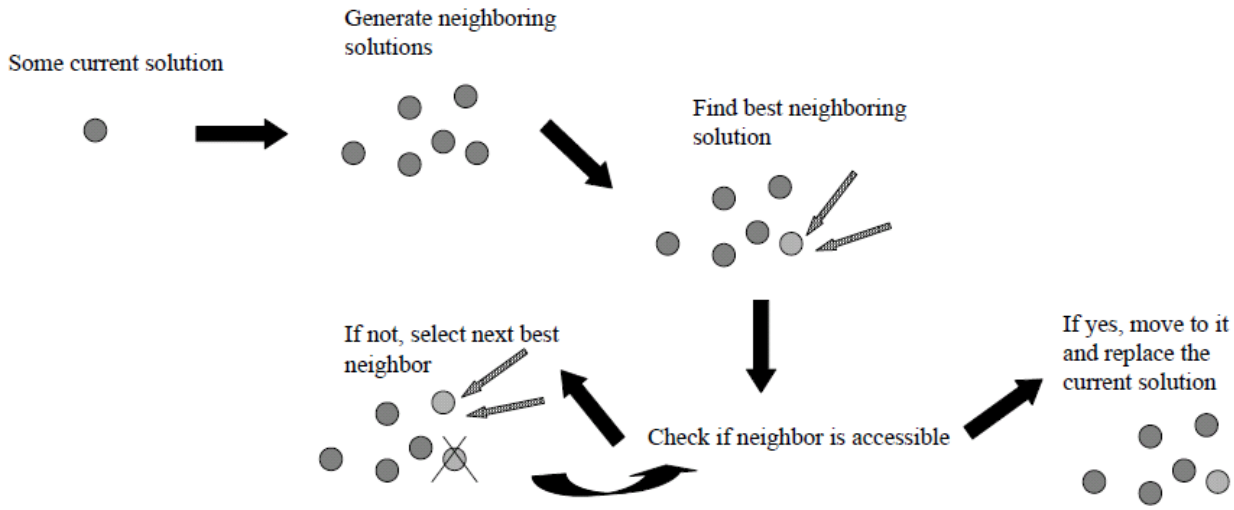
Specific Tabu Search Components

We develop and implement a tabu search based on the basic components outlined in the previous section and Glover's tutorial [14]. A solution consists of arrays that contain the sequence order and the positions of the gaps in the corresponding MSA. The actual MSA is not

stored in memory. The optimality criterion attempts to maximize the most important measure of an MSA -the alignment score. Thus, the quality of an alignment is measured using the final alignment score. Finally, the algorithm terminates after the current best solution has not improved for a specified number of iterations.

There are three types of move strategies that will generate a neighbor from a current solution. The solution representation in Figure 2 displays the first two types of moves. The first type of move is made by swapping pairs of sequences. In Figure 2a) the current solution, is composed of two arrays, s and sg, that contain the sequence order of the MSA and the gap locations, respectively. A possible neighbor, displayed in Figure 2b), is generated by swapping the sequences in positions 3 and 5 as well as the sequences in positions 8 and 10. For this neighbor, the arrays containing the sequence order and gap location are different from

Figure 1: The basic steps of a tabu search.



the arrays in the current solution. The second type of move is made by swapping blocks (two or

more consecutive sequences). For example in Figure 2c, a possible neighbor of the current solution is generated by swapping sequences in positions 2, 3 and 4 with sequences in positions 7, 8 and 9. The third type of move is made by changing the gap positions within a sequence. So, the array containing the sequence order would remain the same and only the array with the gap positions would change.

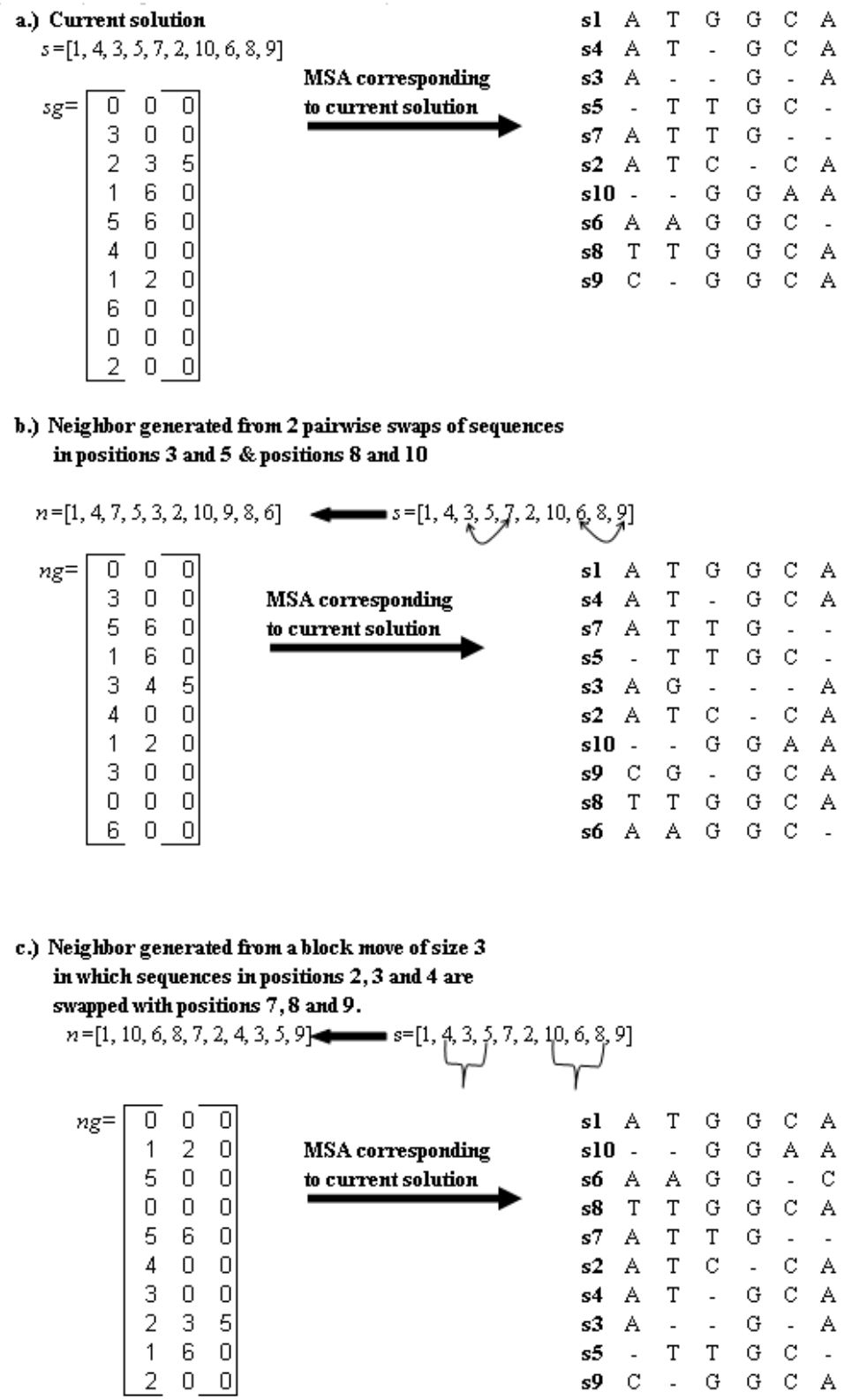
This tabu search, simply progressively aligns all N sequences together in the order specified by a solution. For the MSA in Figure 2a, s1 is aligned with s4 to compose the subalignment, a(s1, s4). Next, s3 is aligned with the subalignment a(s1, s4) to compose the subalignment, a(s1, s4, s3). This progressive alignment continues until the entire MSA, a(s1, s4, s3, s5, s7, s2, s10, s6, s8, s9), is composed.

RESULTS

A multiple sequence generation procedure adapted from Shyu et al. [37] was used to simulate DNA sequences for 4 different groups. This procedure attempts to simulate real biological sequences, with conserved regions that often correspond to important biological functions. Twenty sequences were generated for each of the 4 groups that contained between 18-201 base pairs. For each group of sequences, the tabu search algorithm was run 10 times each.

In Table 1, the highest alignment score in each group from the tabu search was compared with the scores from SAGA, ClustalW, and PRALINE. Also in Table 1, the alignment scores are ranked according to the SP alignment score. A 1 represents the best alignment and 4 represent the worst alignment. As a result of using the sum of pairs scoring function, generally, as the number of sequences or base pairs (BP) increases, the alignment

Figure 2: Solutions representation and moves for the tabu search



score decreases. This decrease in the SP alignment score is the direct result of more gaps with negative gap penalties being introduced into larger MSAs.

MSAs from ClustalW, the most popular commercial alignment program, are used to measure the quality of the alignments from other programs. Quality is measured in terms of the alignment score. Thus, it is assumed that higher alignment scores will yield a higher quality MSA. For all 4 groups, ClustalW yielded MSAs with the highest alignment score. Conversely, in most instances, PRALINE yielded the lowest alignment score. The MSA scores for SAGA were higher than the tabu scores for each of the 4 groups. There was only 1 instance in which the tabu search produced the worst alignment of all the MSA programs.

Table 1: SP alignment scores from ClustalW, SAGA, PRALINE and the tabu search

	No. of Seq, Length	SP Score		No. of Seq, Length	SP Score	
Clus	20 ,18-21	-7.150E+03	1	50,18-21	-1.067E+04	1
SAG	20 ,18-21	-7.150E+03	1	50,18-21	-1.069E+04	2
PRA	20 ,18-21	-7.512E+03	3	50,18-21	-1.074E+04	3
Tabu	20 ,18-21	-7.401E+03	2	50,18-21	-1.088E+04	4
Clus	20, 39-51	-1.500E+04	1	50, 39-51	-3.193E+04	1
SAG	20, 39-51	-1.500E+04	1	50, 39-51	-3.196E+04	2
PRA	20, 39-51	-1.773E+04	3	50, 39-51	-3.298E+04	4
Tabu	20, 39-51	-1.692E+04	2	50, 39-51	-3.215E+04	3
Clus	20, 75-102	-3.776E+04	1	50, 75-102	-2.040E+04	1
SAG	20, 75-102	-3.786E+04	2	50, 75-102	-2.128E+04	2

PRA	20, 75-102	-3.942E+04	3	50, 75-102	-2.279E+04	4
Tabu	20, 75-102	-3.943E+04	4	50, 75-102	-2.278E+04	3
Clus	20, 150-201	-5.901E+04	1	50, 150-201	-7.028E+04	1
SAG	20, 150-201	-5.901E+04	1	50, 150-201	-7.167E+04	2
PRA	20, 150-201	-5.991E+04	3	50, 150-201	-7.391E+04	4
Tabu	20, 150-201	-5.947E+04	2	50, 150-201	-7.385E+04	3

The tabu search is run 10 times for each of the 8 groups of sequences. Each run does not necessarily yield the best alignment score attained using the tabu. The minimum, maximum, average, standard deviation and the percentage of times the best score was reached (for 10 SP scores per group) are displayed in Table 2. The variability in the alignment scores increases as the number of sequences in the group increases. The percentage of times that the best score is reached ranges between 10 and 50 percent. The lack of a diversification procedure explains the widely varying MSA scores and the low percentage of times the best score is reached. It is clear that adding a diverisification procedure would help prevent the tabu from cycling back into local optimal solutions.

Table 2: Measures for the 10 tabu search runs per group using the tabu search

		Tabu Search			
No. of seq, Length	Min SP	Max SP	Avg SP	St Dev	%Max
20, 18-21	-7.52E+03	-7.40E+03	-7.46E+03	8.70E+01	50
20, 39-51	-1.80E+04	-1.69E+04	-1.75E+04	7.79E+02	40
20, 75-102	-4.14E+04	-3.94E+04	-4.04E+04	1.40E+03	40
20, 150-201	-6.14E+04	-5.95E+04	-6.05E+04	1.39E+03	30

CONCLUSION

Tabu search is an effective way to align multiple sequences. This Tabu search does not use a tree to guide the alignment process. One advantage of not using a guide tree is that the tabu search avoids performing pairwise alignments for each pair of sequences in an MSA. When aligning large groups of sequences, making all of the pairwise alignments could become computationally expensive. Another advantage of not using a guide tree (typically produced from a neighbor joining algorithm) is that we can avoid predicting incorrect evolutionary trees. With the addition of a diversification procedure, the tabu search has the potential of producing better alignments that are comparable with ClustalW.

REFERENCES

[1] Abbas, A. and S. Holmes, "Bioinformatics and management science: some common tools and techniques", Operations Research, vol. 52 (2), pp. 165-190, 2004.

[2] Altschul, S.F., R.J. Carroll and D. Lipman, "Weights for data related by a tree", Journal of Molecular Biology, vol. 207, pp. 647-653, 1989.

[3] Altuschul, S.F.,T.L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller and D. Lipman, "Gapped BLAST and PSI BLAST: a new generation of protein database search programs" , Nucleic Acids Research, vol. 25(17), pp. 3389-3402, 1997.

[4] Baeza-Yates, R. A. and G. H. Gonnet, "A new approach to text searching", Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.168-175, 1989.

[5] Brudno, M., S. Malde, A. Poliakov, C. Do, O. Couronne, I. Dubbchak and S. Batzoglou, "Glocal alignment: finding rearrangements during alignment", Bioinformatics, vol. 19,

pp. 154-162, 2003.

[6] Carrillo, H. and D. Lipman, "The multiple sequence alignment problem in biology", SIAM Journal of Applied Mathematics, vol. 48, pp. 1073-1082, 1988.

[7] Davidson, A., "A fast pruning algorithm for optimal sequence alignment", Proceedings of the IEEE 2nd International Symposium on Bioinformatics and Bioengineering Conference, vol 4(6), pp. 49-56, 2001.

[8] Durbin, S., S. Eddy, A. Krogh and G. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, Cambridge, UK, 1998.

[9] http://www.ie.ncsu.edu/fangroup/ps.dir/tabusearch.pdf

[10] Feng, D.F. and R. F. Doolittle, "Progressive sequence alignment as a prerequisite to correct phylogenetic trees", Journal of Molecular Evolution, vol.24(4), pp. 351-360, 1987.

[11] Fitch, W. and J. Farris, "Evolutionary trees with minimum nucleotide replacements from amino acid sequences", Journal of Molecular Evolution, vol. 3, pp. 263-278, 1974.

[12] Gabrani, N. and P. Shankar, "A note on the reconstruction of a binary tree from its traversals", Information Processing Letters, vol. 42(2), pp. 117 -119, 1992.

[13] Gascuel, O, D. Bryant and F. Denis, "Strengths and limitations of the minimum- evolution principle", Systematic Biology, vol. 50(5), pp. 621-627, 2001.

[14] Glover, F., "Tabu search: a tutorial", Interfaces, vol. 20(4), pp. 74-94, 1990.

[15] Gotoh, O., "An improved algorithm for matching biological sequences", Journal of Molecular Biology, vol. 162, pp. 705-708, 1982.

[16] Gotoh, O., "Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments." ,

Journal of Molecular Biology, vol. 264, pp. 823-838, 1996.

[17] Graur, D. and W.H. Li, Fundamentals of Molecular Evolution (2nd ed.), Sinauer Asso- ciates, Sunderland, MA, 2002.

[18] Higgins, D.G. and W. Taylor, Bioinformatics: Sequence Structure and Databanks, Ox- ford University Press, New York, NY, 2000.

[19] Higgins, D.G, and P.M. Sharp, "CLUSTAL: A package for performing multiple sequence alignment on a microcomputer", Gene, vol. 73 ,pp. 237-244, 1988.

[20] Higgins, D.G, J.D. Thompson and T.J. Gibson, "Using CLUSTAL for multiple sequence aligments", Methods Enzymol, vol. 266, pp. 383-402, 1996.

[21] Krogh, A., M. Brown, I.S. Mian, K. Sjolander and D. Haussler, "Hidden markov models in computational biology: applications to protein modeling", Journal of Molecular Biology, vol. 235, pp. 1501-1531, 1994.

[22] Lassmann, T. and E. Sonnhammer, "Kalign: an accurate and fast multiple sequence alignment algorithm", BMC Bioinformatics, vol 6, pp. 298-307 2005

[23] Levenshtein, V., "Binary codes capable of correcting deletions, insertions, and rever- sals", Soviet Physics Doklady, vol.10(8), pp. 707-710, 1966.

[24] Lukashin, A., J. Engelbrecht and S. Brunak, "Multiple alignment using simulated an- nealing: branch point definition in human mRNA splicing", Nucleic Acids Research, vol. 20(10), pp. 2511-2516, 1992.

[25] Zhu, M., G. Hu, Q. Zeng and H. Peng, "Multiple sequence alignment using minimum spanning tree", Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 6, pp.,3352-3356, 2005.

[26] Mount, D. W., Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, 2001.

[27] Needleman, S. B. and C. D. Wunsch, "A general method applicable to the search for similarities in amino acid sequences of two proteins", Journal of Molecular Biology, vol. 48, pp. 443-453, 1970.

[28] Nei, M. and S. Kumar, Molecular Evolution and Phylogenetics , Oxford University Press, New York, NY, 2000.

[29] Notredame, C. and D.G. Higgins, "SAGA: sequence alignment by genetic algorithm", Nucleic Acids Research, vol. 24, pp. 1515-1524, 1996.

[30] Notredame, C., L. Holmes and D.G. Higgins, "COFFEE: an objective function for multiple sequence alignments", Bioinformatics, vol. 14(5), pp. 407-422, 1998.

[31] Ogul, H. and K. Ericyes, "Identifying all local and global alignments between two DNA sequences", International Computing Institute, Ege University, Izmir, Turkey

[32] Rabiner, L.R and B.H. Juang, "An introduction to hidden markov models", IEEE ASSP Magazine, vol. 3(1), pp. 4-16, 1986.

[33] Riaz, T., Y. Wang and K. Li, "Multiple sequence alignment using tabu search", Asia- Pacific Bioinformatics Conference (APBC2004), vol. 29, pp.1-10, 2004.

[34] Rognes, T., "ParAlign: a parallel sequence alignment algorithm for rapid and sensitive database searches", Nucleic Acids Research, vol. 29(7), pp. 1647-1652, 2001.

[35] Schroedl, S., "An improved search algorithm for optimal multiple sequence alignment", Journal of Aritificial Intelligence Research, vol. 23, pp. 587-623, 2005.

[36] Shwartz, A., and L. Pachter, "Multiple alignment by sequence annealing", Bioinformatics, vol. 23(2), pp. 24-29, 2007.

[37] Shyu, C., L. Sheneman and J. Foster, "Multiple sequence alignment with evolutionary computation", Genetic Programming and Evolvable Machines, vol. 5, pp. 121-144, 2004.

[38] Smith, T. F. and M. S. Waterman, "Identification of Common Molecular Subsequences", Journal of Molecular Biology, vol. 147, pp. 195-197, 1981..

[39] Sneath, P. H. A. and R. R. Sokal, Numerical Taxonomy Freeman, San Francisco, CA,.1973.

[40] Thompson, J.D., D.G. Higgins and T.J. Gibson, "CLUSTALW: improving the sensitvity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice, Nucleic Acids Research, vol. 22, pp. 4673-4680, 1994.

[41] http://statgen.ncsu.edu/thorne/bioinf2.html

[42] http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/node3.html

[43] http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html dev/viterbi - - algorithm/s1 pg1.html

[44] Wu, S., U. Manber, E. W. Myers and W. Miller, "An O(NP) sequence comparison algorithm," Information Processing Letters, vol. 35, pp. 17-323, 1990.