2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Development of an efficient sequence alignment algorithm and sequence analysis software

Jin Kim, Jae-Joon Hwang, Dong-Hoi Kim and Saangyong Uhmn
Division of Information and Communication Engineering
Hallym University
1 Ockchundong Chunchon, Kangwondo 200-702
Republic of Korea
Email: {jinkim, director}@hallym.ac.kr, suhmn@ekus.ce.hallym.ac.kr

***Abstract*— Multiple sequence alignment is a useful tool to identify the relationships among protein sequences. Dynamic programming is the most widely used algorithm to obtain multiple sequence alignment with optimal cost. However, dynamic programming cannot be applied to certain cost function due to its drawback and to produce optimal multiple sequence alignment. We proposed sub-alignment refinement algorithm to overcome the problem of dynamic programming and impelmented this algorithm as a module of our MS Windows-based sequence alignment program.**

I. Introduction

Multiple sequence alignment is a useful technique to search the homology in three or more sequences. It has been widely used in the study of molecular structure, function and evolution. The optimal pairwise alignment of any given two sequences can be obtained by dynamic programming introduced by Needleman and Wunsch [1]. Hereafter, many researches were carried out to find efficient algorithms for multiple sequences alignment [2]–[14]. These algorithms can be classified into two categories: the fast heuristic algorithm and exhaustive algorithm. However, the results of heuristic algorithms are not necessarily optimal and the exhaustive algorithms have the limit on the number of sequences because of their complexity.

One of the most important exhaustive algorithms is dynamic programming which is the standard algorithm for multiple sequence alignment. It generates optimal sequence alignment. Lipman et al. [15] implemented a program, Multiple Sequence Alignment(MSA), to align multiple sequences using dynamic programming. Altschul [16] analyzed several types of cost functions for multiple sequence alignment and suggested sum of pairs(SP) as the substitution cost and natural gap cost as the gap cost. In MSA, quasi-natural gap cost is used instead of natural gap cost because of its drawback, which makes MSA fail to generate the optimal alignment in certain cases.

In [17], we analyzed the relation between the costs of alignments with natural gap cost and quasi-matural gap cost and suggested sub-alignment refinement method to overcome the drawback of MSA.

This paper is organized as follows. In section 2 and 3, we states the definition of multiple sequence alignment and our proposed method for the completeness of the paper. We give a brief explanation about our implementation of dynamic programming with the proposed method for MS Windows in section 4. We give a conclusion in section 5 with the future work.

II. Multiple Sequence Alignment

A. Definition

Before we give the definition of multiple sequence alignment, we need the following terms.

- *Alphabet* Σ' is a finite set of letter(Σ) and a null('-').
- A *sequence* is a finite string over Σ'.
- A *gap* is a finite consecutive nulls.
- input sequences, $S_1, S_2, \cdots, S_k$, are k sequences of the length $n_1, n_2, \cdots, n_k$ respectively ($S_1 = s_{11}s_{12}\cdots s_{1n_1}$, $S_2 = s_{21}s_{22}\cdots s_{2n_2}$, $\cdots$, $S_k = s_{k1}s_{k2}\cdots s_{kn_k}$) over Σ and $s_{ij} \in \Sigma$ is the j^{th} element of i^{th} sequence($1 \le i \le k$, $1 \le j \le n_i$). The alignment of the input sequences is a set of k padded sequences $S'_1 = s'_{11}s'_{12}\cdots s'_{1l}$, $S'_2 = s'_{21}s'_{22}\cdots s'_{2l}$, $\cdots$, $S'_k = s'_{k1}s'_{k2}\cdots s'_{kl}$ of equal length l. We can obtain the input sequences by removing the nulls from S'_i.

Definition 1: Multiple sequence alignment
For the given set of input sequences, $S_1, S_2, \cdots, S_k$, over Σ', find an alignment $S'_1, S'_2, \cdots, S'_k$ of equal length l with optimal cost based on the given cost functions.

This problem is known to be *NP-complete* so that it is impractical to find the optimal solution in polynomal time.

B. Cost function

A cost function should be explicitly defined to represent the quality of sequence alignment. In [16], Alstchul analyzed several cost functions for sequence alignment and divided the cost functions into two categories: substitution cost and gap cost.

1) Substitution cost: Substitution cost is the cost for aligning sequences, which includes cost for aligning letters which represent amino acids with nulls. The sum of pairs(SP) substitution cost is used in MSA. The SP substitution cost can be obtained by summing the costs for aligning $n(n-1)/2$ pairs for n sequences. The substitution cost of an alignment

TABLE I

NATURAL AND QUASI-NATURAL GAP COST

Gap cost	sequence	
	X---X XXX-X XXXXX	X---X XX-XX XXXXX
natural gap cost	3	3
quasi-natural gap cost	3	4

A, $C_1(A)$, can be obtained as follows:

$$C_1(A) = \sum_{i=1}^{n-1} \sum_{j=i}^{n} Cost(S'_i, S'_j) \cdot weight(i,j) \qquad (1)$$

where $Cost(S'_i, S'_j)$ is the cost for aligning a pair of sequences, S'_i and S'_j, and $weight(i,j)$ is its weight. In MSA, PAM250 weight matrix is used to calculate substitution cost [18].

2) Gap cost: The gap is a string of consecutive nulls in a sequence aligned with letters in the other sequences. Gap cost is assessed separately from null cost. The null cost is the substitution cost for aligning individual letter with a null. If there are nulls at the identical location in two sequences, S'_i and S'_j, those nulls can be removed. Altschul proposed the natural gap cost which is clearly related to the substitution cost. But to apply natural gap cost, we have to keep lots of information about alignment history. As a solution to this problem, Altschul proposed quasi-natural gap cost which is similar to natural gap cost except for some special cases and massively reduces the amount of information kept in a cell($2^n - 1$ for n sequences), which makes it more practical than natural gap cost. The gap cost of an alignment A, $C_2(A)$, can be obtained by the following expression:

$$C_2(A) = k \cdot g \qquad (2)$$

where k is the number of added gaps in A and g is the gap penalty.

We now can obtain the total cost of an alignment A, $C(A)$, by summing these two costs of (1) and (2), $C(A) = C_1(A) + C_2(A)$. Let $S = A_1, A_2, \cdots, A_m$ be the set of all possible sequence alignments with the same set of input sequences. The multiple sequence alignment problem is to find the alignment A_i whose cost $C(A_i)$ is the smallest among them.

C. Natural gap cost and quasi-natural gap cost

MSA uses quasi-natural gap cost instead of natural gap cost so that we define $Opt_{quasi-natural}$ as the cost of the alignment by MSA and $Opt_{natural}$ as that of the alignment with natural gap cost. In Table I, we show the difference between the natural gap cost and the quasi-natural gap cost where X is any character except '-'. As seen in the first alignment in Table I, if there is no completely nested gap, $Opt_{quasi-natural}$ and $Opt_{natural}$ have the same value. However, if there is a completely nested gap as in the second alignment, quasi-natural cost counts one more gap than natural gap cost does, which makes $Opt_{quasi-natural}$ is greater than $Opt_{natural}$ by the number of added gaps times gap penalty.

III. ALGORITHM FOR $Opt_{natural}$

We can obtain the alignment with $Opt_{quasi-natural}$ by MSA because quasi-natural gap cost is used in MSA. However, our goal is to obtain one with $Opt_{natural}$ using natural gap cost. Now, we present Sub-alignment Refinement Algorithm(SRA) which chooses the partial alignment from the result by MSA with the possibility of completely nested gaps and tries to improve this alignment with natural gap cost to obtain the alignment with $Opt_{natural}$.

A. Realingment

We can get the optimal or near optimal alignment with MSA. We observe that two alignments in Table I are identical except those regions with nulls. So if we can align those regions optimally, we can make the whole alignment be optimal easily. To apply our algorithm, the regions must satisfy the following conditions:

1) two or more sequences must contain gaps.
2) the difference between the maximum and the minimum number of nulls must be 2 or more.

These are the least conditions for the completely nested gaps. If these conditions are met, we can realign the nulls in that region to generate sub-alignment with completely nested gaps and calculate the cost with natural gap cost. If we have a sub-alignment with smaller cost, we can substitute them for that region so that we can have the whole alignment with the smaller cost. If we have a sub-alignment with the smallest cost possible, we can make the whole alignment be optimal. In Fig.1, we present the proposed algorithm formally. In [17], you can find the the relation between $Opt_{quasi-natural}$ and $Opt_{natural}$ and theoritical bases of our algorithm.

B. Analysis of algorithm

In Fig.1, we present the algorithm. The time complexity of the algorithm depends on that of [Step 2]. In [Step 2], we compute the cost of a sub-alignment in $O(ln^2)$ and $O(Kln^2)$ for K sub-alignments where n is the number of sequences, l their length in a sub-alignment and K is the number of the generated sub-alignments,

$$K = \prod_{i=1}^{n} (l - G_i + 1) \qquad (3)$$

where G_i is the length of gap in a sequence S_i. And it has the space complexity of $O(ln)$ in [Step 2] to store those sequences.

IV. THE IMPLEMENTATION

We implemented a software which can be easily used by endusers. Most of the users for sequence alignments are used to window based softwares. The developed software was written by Visual C++. The proposed algorithm in this paper and other well known algorithms for multiple sequence alignment including MSA and Clustal W are inplemented in this software. Users can apply various algorithms to their sequences and compare the results from the algorithms. Also

Step 1 determine the sub-alignments with the possible improvement based the output of MSA, which must satisfy the above two conditions.
Step 2 apply realignment and recalculation to this sub-alignment
Step 3 if new sub-alignment has $Opt_{natural}$, substitute them for the previous sub-alignment

Fig. 1. The proposed algorithm

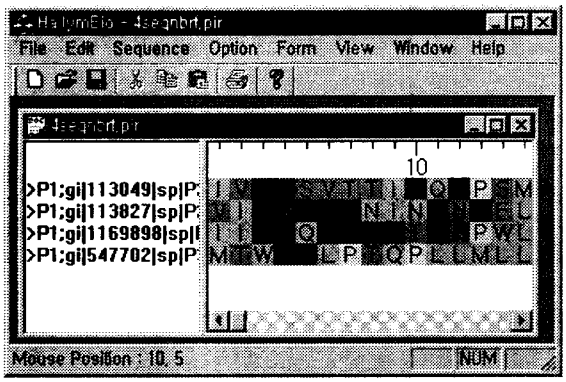

Fig. 2. Input sequence window

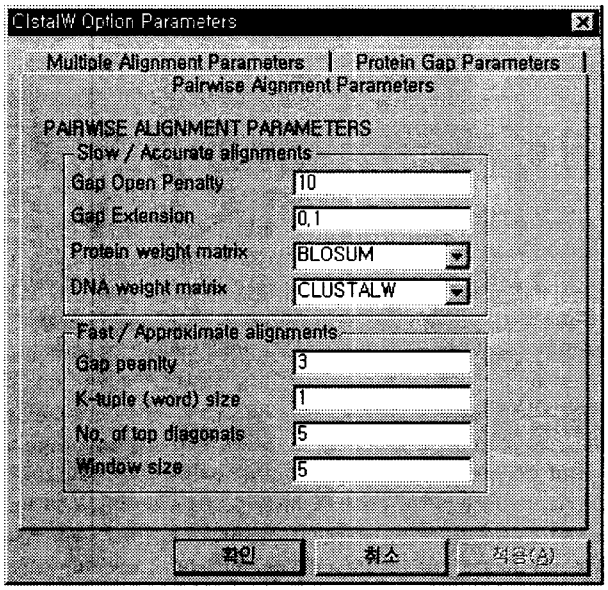

Fig. 3. Clustal W parameter window

there are many useful aspects in this software. Users can directly obtain useful biological information from the biological database in the Internet. They can use blast program to search the similar seqeuences in the biological databases. Their own local sequence databases can be generated and used by our software. We are going to implement our algorithm with other widely used ones for multiple sequence alignement. As the first, we implemented our one with Clustal W. We give The snapshots and brief explanations of the implementation in the rest of this paper.

In Fig.2, we can see the input sequences in colors. We can edit the sequences such that deleting and inserting at any portion of sequences are possible.

As stated above, we now have MSA with SRA and Clustal W in this package so that we can get the alignment result by any of those. To align the sequences, we can set some parameters. In Fig.3, we see a snapshot of the Clustal W parameter window. In Fig.4 and Fig.5, the snapshots of the result windows of MSA and Clustal W are provided. As you can see from those figures, we can apply MSA with SRA or Clustal W to the same set of input sequences without changing the platform

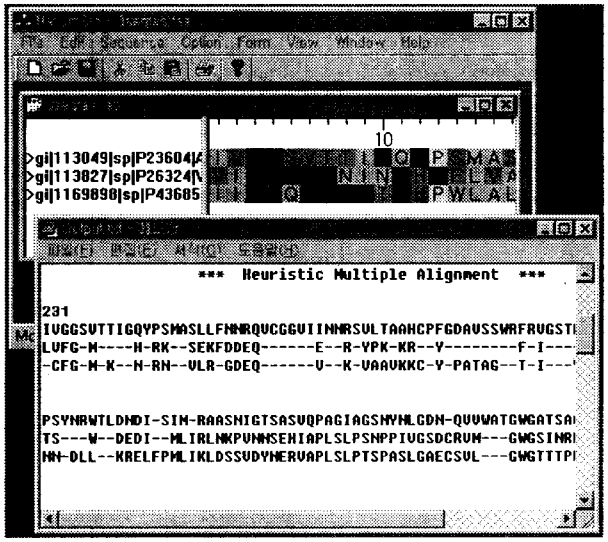

Fig. 4. Result of MSA with SRA

V. Conclusion

First, we discussed the optimization technique for multiple sequence alignment. The dynamic programming is the mose widely used algorithm in this field. However it has a drawback that it can not use the natural gap cost. Even though many researchers proposed algorithms to solve this problem, none of them explained the relationship between the optimal alignment, $Opt_{natural}$, and $Opt_{quasi-natural}$ by dynamic programming. In this paper, we provided the theoretical basis for the lower bound of $Opt_{natural}$ and proposed the algorithm to improve the alignment by the dynamic programming.

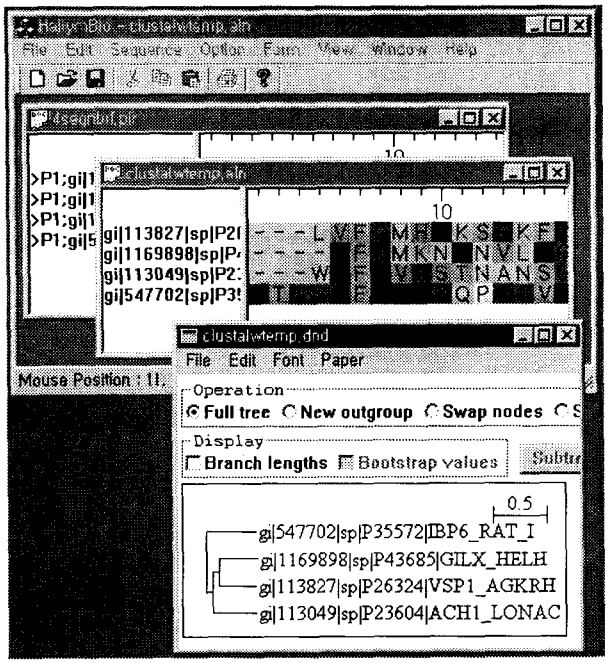

Fig. 5. Result of Clustal W

Second, we present our implementation of MSA with SRA. In the implementation, we integrated other multiple sequence alignemtn algorithms with ours. By integrating several widely used multiple sequence alignment algorithms in one package, the user can have the benefit of the ease of use of those algorithms without changing alignment softwares or websites.

ACKNOWLEDGMENT

This study was supported by a grant of the International Mobile Telecommunications 2000 R&D Project, Ministry of Information & Communication, Republic of Korea.

REFERENCES

[1] S. B. NEEDLEMAN and C. D. WUNCH, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Molec. Biol.*, Vol. 48, pp. 443–453, 1970.

[2] A. F. ALTSCHUL and D. J. LIPMAN, "Threes, stars, and multiple biological sequence alignment," *SIAM J. appl. Math.*, Vol. 49 pp. 197–209, 1989.

[3] S. C. C. CHAN, A. K. WONG, and D. K. Y. CHIU, "A survey of multiple sequence comparison methods," *Bull. Math. Bio.*, Vol. 43, pp. 563–598, 1992

[4] D. F. FENG and R. F. DOOLITTLE, "Progressive sequence alignment as a prerequisite to correct phylogenetic trees," *J. Molec. Evol.*, 25:351–360, 1987.

[5] D. F. FENG, M. S. JOHNSON, and R. F. DOOLITTLE, "Aligning amino acid sequences: comparison of commonly used methods," *J. Molec. Evol.*, Vol. 21, pp. 112–125, 1982.

[6] J. W. FICKET, "Fast optimal alignment," *Nucl. Acids Res.*, Vol. 12 pp. 175–180, 1984.

[7] J. KIM and S. PRAMANIK, "An efficient method for multiple sequence alignment," *In Proc. Second International Conference on Intelligent Systems for Molecular Biology*, 1994, pp. xxx-xxx.

[8] J. KIM, S. PRAMANIK, and M. J. CHUNG "Multiple sequence alignment using simulated annealing," *CABIOS*, Vol.10, pp. 419–426, 1994.

[9] H. M. MARTINEZ, "A flexible multiple sequence alignment program," *Nucl. Acids. Res.*, Vol. 16, pp. 1683–1691, 1988.

[10] M. MURATA, J. S. RICHARDSON, and J. L. SUSSMAN, "Simultaneous comparison of three protein sequences," *In Proc. Natl. Acad. Sci. USA.*, Vol. 82, pp. 3073–3077, 1985.

[11] C. NOTREDAME and D. HIGGINS, "SAGA:sequence alignment by genetic algorithm," *Nucl. Acids. Res.*, Vol. 24, No. 8, pp. 1515–1524, 1996

[12] D. SANKOFF, *Simultaneous comparison of three or more sequences related by a tree.* Addison-Wesley, Reading, MA, 1983.

[13] D. SANKOFF and J. B. KRUSKAL, *Time Warps, String Edits and Macromolecules: The theory and practice of Sequence Comparison.* Addison-Wesley, Reading, MA, 1983.

[14] W. R. TAYLOR Multiple sequence alignment by a pairwise algorithm, *CABIOS*, Vol.3, pp.81–87, 1987.

[15] D. J. LIPMAN, S. F. ALTSCHUL, and J. D. KECECIOGLU, "A tool for multiple sequence alignment," *Proc. Natl. Acad. Sci. USA.*, Vol.86, pp. 4412–4415, 1989.

[16] S. F. ALTSCHUL, "Gap costs for multiple sequence alignment," *J Theor. Biol.*, Vol.138 pp. 297–309, 1989.

[17] J. KIM, and S. UHMN, "An efficient method for multiple sequence alignment using subalignment refinement," *In Proc. METMBS*, 2003, pp. xxx-xxx.

[18] M. O. DAYHOFF, A model of evolutionary change in proteins. matrices for detecting distance relationships, *In Atlas of Protein sequence an Structure*, Vol. 5 suppl.3, pp.354–352. Dayhoff. M. O.(ed) Washington. DC: National Biomedical Research Foundation, 1978.