Journal of Bioinformatics and Computational Biology
Vol. 1, No. 2 (2003) 267–287
© Imperial College Press

Imperial College Press
www.icpress.co.uk

CONSTRAINED MULTIPLE SEQUENCE ALIGNMENT TOOL DEVELOPMENT AND ITS APPLICATION TO RNase FAMILY ALIGNMENT*

CHUAN YI TANG[†]

Department of Computer Science, National Tsing Hua University,
Hsinchu 300, Taiwan, ROC
[†]*cytang@cs.nthu.edu.tw*

CHIN LUNG LU

Department of Biological Science and Technology, National Chiao Tung University,
Hsinchu 300, Taiwan, ROC

MARGARET DAH-TSYR CHANG

Department of Life Science, National Tsing Hua University,
Hsinchu 300, Taiwan, ROC

YIN-TE TSAI

Department of Computer Science and Information Management,
Providence University, Shalu, Taichung Hsien 433, Taiwan, ROC

YUH-JU SUN

Department of Life Science, National Tsing Hua University,
Hsinchu 300, Taiwan, ROC

KUN-MAO CHAO

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei 106, Taiwan, ROC

JIA-MING CHANG and YU-HAN CHIOU

Department of Computer Science, National Tsing Hua University,
Hsinchu 300, Taiwan, ROC

CHIA-MAO WU, HAO-TENG CHANG and WEI-I CHOU

Department of Life Science, National Tsing Hua University,
Hsinchu 300, Taiwan, ROC

*A preliminary version of this paper appears in the Proceedings of the First IEEE Computer Society Bioinformatics Conference (CSB 2002). This research was supported partly by VTY89-P4-33, NSC89-2213-E-259-010, NSC91-2321-B-007-002, NSC91-3112-B-007-004 and MOE Program for Promoting Academic Excellence of Universities under the grant number 89-B-FA04-1-4.
[†]Corresponding author.

Received 15 September 2002
Revised 7 December 2002
Accepted 15 January 2003

In this paper, we design a heuristic algorithm of computing a constrained multiple sequence alignment (CMSA for short) for guaranteeing that the generated alignment satisfies the user-specified constraints that some particular residues should be aligned together. If the number of residues needed to be aligned together is a constant α, then the time-complexity of our CMSA algorithm for aligning K sequences is $\mathcal{O}(\alpha K n^4)$, where n is the maximum of the lengths of sequences. In addition, we have built up such a CMSA software system and made several experiments on the RNase sequences, which mainly function in catalyzing the degradation of RNA molecules. The resulting alignments illustrate the practicability of our method.

Keywords: Computational biology; constrained multiple sequence alignment; RNases.

1. Introduction

Multiple sequence alignment (MSA for short) is one of the most important problems in computational biology.[5,6] The sum-of-pairs (SP for short) score is widely used criterion for selecting the optimal alignment. This kind of MSA problem, called sum-of-pairs MSA (SPMSA for short) problem, can be solved by extending the dynamic programming algorithm of Needleman and Wunsch for aligning two sequences.[20] In the worst case, however, it needs to take $O(2^K n^K)$ time to align K sequences of length n. This exponential time limits the dynamic programming technique to align only a small number of short sequences. Actually, the SPMSA problem has been shown to be NP-complete,[4,32] which means that it seems to be impossible to design an efficient algorithm to find the mathematically optimal alignment. Hence, some approximate and heuristic methods are introduced to overcome this problem.

For the approximate methods, Gusfield[11] first proposed a polynomial-time approximation algorithm with performance ratio of $2-\frac{2}{K}$. Then Pevzner[22] improved the performance ratio to $2-\frac{3}{K}$. Recently, Bafna, Lawler and Pevzner[2] further improved the performance ratio to $2-\frac{l}{K}$ for any fixed l. It is worth mentioning that Li, Ma and Wang[17] have given a polynomial time approximation scheme (PTAS for short) for finding a multiple sequence alignment within a constant band, which is often useful in many practical cases. For the heuristic methods, the most widely used heuristic methods are the so-called progressive strategies.[7,9,13,28,29]

Usually, biologists have the knowledge of their datasets concerning the structures, active site residues, intramolecular disulfide bonds, substrate binding sites and enzyme activities. For example, all living organisms contain ribonucleases (RNases) which mainly function in the ribonucleic acids (RNA) processing such as RNA maturation and turnover by catalyzing the degradation of RNAs. Many ribonucleases including bovine and human pancreatic RNaseAs have been isolated and characterized in terms of their amino acid sequences, coding genes, three-dimensional structures and biological functions. As compared to bovine pancreatic

RNaseA, the major structural features of all RNases contain three conserved His12, Lys41 and His119 active site residues and four disulfide bonds. Since the RNases with solved 3D structures all show very high homology among the catalytic domains and disulfide linkages, we would expect that their primary sequence comparisons to be matched very well. In other words, their alignment should place His12 (respectively, Lys41 and His119) of bovine pancreatic RNase and other His (respectively, Lys and His) residues in the same column. The pairwise alignment of most RNases to the bovine pancreatic RNaseA shows perfect matches of the three key amino acid residues. According to our test cases, however, multiple sequence alignment of more RNases employing the existing computer programs always generates mismatches among the important residues (see Sec. 5 for the details). To solve the problem, the biologists need a novel multiple sequence alignment adapting all known information about the structures, active site residues, intramolecular disulfide bonds, substrate binding sites and enzyme activities about a particular subject.

In this paper, we design a heuristic method of computing a constrained multiple sequence alignment (CMSA for short) for guaranteeing that the generated alignment satisfies the user-specified constraints that some particular residues should be aligned together. Our strategy is first to design the constrained pairwise sequence alignment, then create a guide tree, called Kruskal merging order tree, based on the Kruskal minimum spanning tree of the sequences, and finally according to the branching order of the Kruskal merging order tree, align the sequences progressively using the constrained pairwise sequence alignment. If the number of residues needed to be aligned together is a constant α, then the time-complexity of our CMSA algorithm for aligning K sequences is $\mathcal{O}(\alpha K n^4)$, where n is the maximum of the lengths of sequences. In addition, we have built up such a CMSA software system and made several experiments on the RNase sequences. The resulting alignments illustrate the practicability of our method. Note that Tang *et al.* have designed an $\mathcal{O}(\alpha n^6)$ time algorithm for solving the constrained pairwise sequence alignment.[27]

It is worth mentioning that Myers *et al.* have proposed another version of the CMSA for the different applications.[19] In their version, they considered a different kind of constraints to restrict the alignment such that certain sequence positions should appear relative to others. For instance, they defined a kind of constraint, denoted by $i_s \preceq j_t$, to assert that in the alignment, the sth residue of the ith sequence should occur in the same column as or before the column containing the tth residue of the jth sequence. By defining a cyclic chain of K $\preceq$-constraints, they are able to restrict the alignment such that the sequence positions in distinct sequences should be aligned in the same column. However, this kind of CMSA is different from the one we study in this paper. This is because that the constraints considered by Myers *et al.* want the sequence positions, which are known in advance, to be aligned together, while the constraints we study here require the residues of the same type (e.g. His or Lys), whose sequence positions are unknown in advance, being aligned together. Given a collection of C $\preceq$-constraints over K sequences whose total length is N, the algorithm of Myers *et al.* takes $\mathcal{O}(K(N^2 + KC))$ time.

The rest of this paper is organized as follows. In Sec. 2, we give a formal definition of the problem and introduce some definitions used in this paper. Then we describe the algorithms of the constrained pairwise and multiple sequence alignments in Secs. 3 and 4, respectively. In Sec. 5, we describe the motivation of developing the CMSA tool in more details and experiment it on the RNase sequences for illustrating the practicability of our method. Finally, we give concluding remarks in Sec. 6.

2. Preliminaries

Let $S = \{S_1, S_2, \ldots, S_K\}$ be the set of K sequences over the alphabet Σ, where sequence $S_i = s_1^i s_2^i \cdots s_{n_i}^i$ has length n_i for each $1 \le i \le K$. For each S_i of S, we let $S_i[x, y] = s_x^i s_{x+1}^i \cdots s_y^i$, where $1 \le x < y \le n_i$. Then a *multiple sequence alignment* (MSA) of S is a rectangular matrix consisting of K rows of characters of $\Sigma \cup \{-\}$ such that no column consists entirely of dashes and removing dashes from row i leaves S_i for any $1 \le i \le K$. The *sum-of-pairs score* (SP score) of an MSA is defined to be the sum of the scores of all columns, where the score of each column is the sum of the scores of all distinct pairs of characters in the column. In practice, the score of the pair of two dashes is usually set to zero. Then the problem of finding an MSA of S with the optimal SP score is the so-called *sum-of-pairs MSA* (SPMSA) *problem*.[5,6] Suppose that $P = p_1 p_2 \cdots p_\alpha$ is a common subsequence of $S_1, S_2, \ldots, S_K$ (i.e. P can be obtained from each S_i by removal of some non-consecutive characters). Then the *constrained multiple sequence alignment* of S *with respect to* P is an MSA $\mathcal{A}$ with the constraints that there are α columns in $\mathcal{A}$, say columns $c_1, c_2, \ldots, c_\alpha$ with $c_1 < c_2 < \cdots < c_\alpha$, such that the characters of column c_i, $1 \le i \le \alpha$, are all equal to p_i. The so-called *constrained multiple sequence alignment problem* is to find a constrained MSA with the optimal SP score. In this paper, we adopt that the optimal alignment is the one with maximum SP score.

Given a connected and undirected graph $G = (V, E)$ in which each edge of E is associated with a positive weight, the *minimum spanning tree* (MST) of G is a tree T consisting of all the vertices of V such that the sum of the weights of all edges in T is minimum. Kruskal[15] gave an algorithm of $\mathcal{O}(|V|^2 \log |V|)$ time for computing an MST of G, where $|V|$ denotes the size of V. Here, we call the MST constructed by Kruskal's algorithm as a *Kruskal MST*.

3. Constrained Pairwise Sequence Alignment

In this section, we will consider the problem of finding a constrained alignment of two sequences with the optimal score (i.e. the so-called constrained pairwise sequence alignment problem). Recall that Tang *et al.* gave an $\mathcal{O}(\alpha n^6)$ time algorithm for solving this problem using the dynamic programming technique.[27] In the following, we improve their result to $\mathcal{O}(\alpha n^4)$ time.

For any two characters s and s' over $\Sigma \cup \{-\}$, let $f(s, s')$ denote the score of aligning s with s'. Usually, $f(-,-) = 0$. Let $S_1 = s_1^1 s_2^1 \cdots s_{n_1}^1$ and $S_2 = s_1^2 s_2^2 \cdots s_{n_2}^2$

be any two sequences and $P = p_1p_2 \cdots p_\alpha$ be a given common subsequence of S_1 and S_2. In the following, we define some useful notation.

- Let $\mathcal{M}_C(S_1, S_2)$ be the optimal score of the constrained sequence alignment of S_1 and S_2 with respect to P.
- For $1 \le x \le i \le n_1$ and $1 \le y \le j \le n_2$, let $\mathcal{M}_0(x, i; y, j)$ be the optimal score of globally aligning subsequences $S_1[x, i]$ and $S_2[y, j]$ without any constraint. For convenience, we let $\mathcal{M}_0(x, i; y, j) = 0$ if $x > i$ or $y > j$.
- For $1 \le i \le n_1$, $1 \le j \le n_2$ and $1 \le k \le \alpha$, let $\mathcal{M}_k(i, j)$ be the optimal score of the constrained sequence alignment $\mathcal{A}$ of $S_1[1, i]$ and $S_2[1, j]$ with respect to $P[1, k]$, where $P[1, k] = p_1p_2 \cdots p_k$, such that s_i^1 is aligned with s_j^2 and $s_i^1 = s_j^2 = p_k$. That is, there are k columns in $\mathcal{A}$, say columns $c_1, c_2, \ldots, c_k$ with $c_1 < c_2 < \cdots < c_k$, such that the characters of column c_h, $1 \le h \le k$, are all equal to p_h, and c_k is the last column of $\mathcal{A}$ consisting of s_i^1 and s_j^2. Note that if the constrained sequence alignment $\mathcal{A}$ does not exist, we let $\mathcal{M}_k(i, j) = -\infty$.

By definition, we have the following lemma immediately.

Lemma 1. *For all* $1 \le i \le n_1$ *and* $1 \le j \le n_2$, *if* $s_i^1 = s_j^2 = p_1$, *then we have* $\mathcal{M}_1(i, j) = \mathcal{M}_0(1, i-1; 1, j-1) + f(s_i^1, s_j^2)$; *otherwise,* $\mathcal{M}_1(i, j) = -\infty$.

Lemma 2. *For all* $1 \le i \le n_1$, $1 \le j \le n_2$ *and* $2 \le k \le \alpha$, *if* $s_i^1 = s_j^2 = p_k$, *then we have* $\mathcal{M}_k(i, j) = \max_{1 \le x < i, 1 \le y < j}\{\mathcal{M}_{k-1}(x, y) + \mathcal{M}_0(x+1, i-1; y+1, j-1) + f(s_i^1, s_j^2)\}$; *otherwise,* $\mathcal{M}_k(i, j) = -\infty$.

Proof. By definition, if the condition of $s_i^1 = s_j^2 = p_k$ does not hold, then we have $\mathcal{M}_k(i, j) = -\infty$. In the following, we assume $s_i^1 = s_j^2 = p_k$ and let $\mathcal{A}$ be an optimal constrained sequence alignment of $S_1[1, i]$ and $S_2[1, j]$ with respect to $P[1, k]$. In other words, we can find k columns in $\mathcal{A}$, say columns $c_1, c_2, \ldots, c_k$ with $c_1 < c_2 < \cdots < c_k$, such that the characters of column c_h, $1 \le h \le k$, are all equal to p_h, where c_k is the last column of $\mathcal{A}$ consisting of s_i^1 and s_j^2. In this case, we can decompose $\mathcal{A}$ into three sub-matrixes: (1) the sub-matrix, denoted by $\mathcal{A}_1$, consisting only of the last column c_k, (2) the sub-matrix, denoted by $\mathcal{A}_2$, consisting of those columns before and including column c_{k-1}, and (3) the sub-matrix, denoted by $\mathcal{A}_3$, consisting of the remaining columns. Assume that the column c_{k-1} consists of s_x^1 and s_y^2, where $1 \le x < i$ and $1 \le y < j$. Then it is clear that $\mathcal{A}_2$ corresponds to an optimal constrained sequence alignment of $S_1[1, x]$ and $S_2[1, y]$ with respect to $P[1, k-1]$, and $\mathcal{A}_3$ corresponds to an optimal alignment of $S_1[x+1, i-1]$ and $S_2[y+1, j-1]$ without any constraint. That is, $\mathcal{M}_k(i, j) = \mathcal{M}_{k-1}(x, y) + \mathcal{M}_0$ $(x+1, i-1; y+1, j-1) + f(s_i^1, s_j^2)$. Since $\mathcal{M}_k(i, j)$ is maximum, it is not hard to see that $\mathcal{M}_k(i, j) = \max_{1 \le x < i, 1 \le y < j}\{\mathcal{M}_{k-1}(x, y) + \mathcal{M}_0(x+1, i-1; y+1, j-1) + f(s_i^1, s_j^2)\}$. □

Lemma 3. $\mathcal{M}_C(S_1, S_2) = \max_{1 \le i \le n_1, 1 \le j \le n_2}\{\mathcal{M}_\alpha(i, j) + \mathcal{M}_0(i+1, n_1; j+1, n_2)\}$.

The rest of this paper is organized as follows. In Sec. 2, we give a formal definition of the problem and introduce some definitions used in this paper. Then we describe the algorithms of the constrained pairwise and multiple sequence alignments in Secs. 3 and 4, respectively. In Sec. 5, we describe the motivation of developing the CMSA tool in more details and experiment it on the RNase sequences for illustrating the practicability of our method. Finally, we give concluding remarks in Sec. 6.

2. Preliminaries

Let $S = \{S_1, S_2, \ldots, S_K\}$ be the set of K sequences over the alphabet Σ, where sequence $S_i = s_1^i s_2^i \cdots s_{n_i}^i$ has length n_i for each $1 \leq i \leq K$. For each S_i of S, we let $S_i[x, y] = s_x^i s_{x+1}^i \cdots s_y^i$, where $1 \leq x < y \leq n_i$. Then a *multiple sequence alignment* (MSA) of S is a rectangular matrix consisting of K rows of characters of $\Sigma \cup \{-\}$ such that no column consists entirely of dashes and removing dashes from row i leaves S_i for any $1 \leq i \leq K$. The *sum-of-pairs score* (SP score) of an MSA is defined to be the sum of the scores of all columns, where the score of each column is the sum of the scores of all distinct pairs of characters in the column. In practice, the score of the pair of two dashes is usually set to zero. Then the problem of finding an MSA of S with the optimal SP score is the so-called *sum-of-pairs MSA* (SPMSA) *problem*.[5,6] Suppose that $P = p_1 p_2 \cdots p_\alpha$ is a common subsequence of $S_1, S_2, \ldots, S_K$ (i.e. P can be obtained from each S_i by removal of some non-consecutive characters). Then the *constrained multiple sequence alignment* of S *with respect to* P is an MSA $\mathcal{A}$ with the constraints that there are α columns in $\mathcal{A}$, say columns $c_1, c_2, \ldots, c_\alpha$ with $c_1 < c_2 < \cdots < c_\alpha$, such that the characters of column c_i, $1 \leq i \leq \alpha$, are all equal to p_i. The so-called *constrained multiple sequence alignment problem* is to find a constrained MSA with the optimal SP score. In this paper, we adopt that the optimal alignment is the one with maximum SP score.

Given a connected and undirected graph $G = (V, E)$ in which each edge of E is associated with a positive weight, the *minimum spanning tree* (MST) of G is a tree T consisting of all the vertices of V such that the sum of the weights of all edges in T is minimum. Kruskal[15] gave an algorithm of $\mathcal{O}(|V|^2 \log |V|)$ time for computing an MST of G, where $|V|$ denotes the size of V. Here, we call the MST constructed by Kruskal's algorithm as a *Kruskal MST*.

3. Constrained Pairwise Sequence Alignment

In this section, we will consider the problem of finding a constrained alignment of two sequences with the optimal score (i.e. the so-called constrained pairwise sequence alignment problem). Recall that Tang *et al.* gave an $\mathcal{O}(\alpha n^6)$ time algorithm for solving this problem using the dynamic programming technique.[27] In the following, we improve their result to $\mathcal{O}(\alpha n^4)$ time.

For any two characters s and s' over $\Sigma \cup \{-\}$, let $f(s, s')$ denote the score of aligning s with s'. Usually, $f(-, -) = 0$. Let $S_1 = s_1^1 s_2^1 \cdots s_{n_1}^1$ and $S_2 = s_1^2 s_2^2 \cdots s_{n_2}^2$

Proof. Let $\mathcal{A}$ be an optimal constrained sequence alignment of S_1 and S_2 with respect to P. By definition, we can find α columns in $\mathcal{A}$, say columns $c_1, c_2, \ldots, c_\alpha$ with $c_1 < c_2 < \cdots < c_\alpha$, such that the characters of column c_k, $1 \le k \le \alpha$, are all equal to p_k. Assume that column c_k consists of s_i^1 and s_j^2, where $1 \le i \le n_1$ and $1 \le j \le n_2$. Then we can decompose $\mathcal{A}$ into two sub-matrixes: (1) the sub-matrix, denoted by $\mathcal{A}_1$, consisting of those columns before and including column c_α, and (2) the sub-matrix, denoted by $\mathcal{A}_2$, consisting of the remaining columns. Clearly, $\mathcal{A}_1$ corresponds to an optimal constrained sequence alignment of S_1 $[1, i]$ and $S_2[1, j]$ with respect to P, and $\mathcal{A}_2$ corresponds to an optimal alignment of $S_1[i+1, n_1]$ and $S_2[j+1, n_2]$ without any constraint. Hence, $\mathcal{M}_C(S_1, S_2) = \mathcal{M}_\alpha(i, j) + \mathcal{M}_0(i+1, n_1; j+1, n_2)$. Since $\mathcal{M}_C(S_1, S_2)$ is maximum, it is not hard to see that $\mathcal{M}_C(S_1, S_2) = \max_{1 \le i \le n_1, 1 \le j \le n_2}\{\mathcal{M}_\alpha(i, j) + \mathcal{M}_0(i+1, n_1; j+1, n_2)\}$. □

According to Lemmas 1, 2 and 3, we are able to design a dynamic programming algorithm, called as Algorithm C2SA, for solving the constrained pairwise sequence alignment problem. Before it, we describe a pre-processing algorithm to efficiently compute $\mathcal{M}_0(x, i; y, j)$ for all $1 \le x \le i \le n_1$ and $1 \le y \le j \le n_2$. The basic idea of this pre-processing algorithm is as follows. For all pairs of a suffix $S_1[x, n_1]$ of S_1 and a suffix $S_2[y, n_2]$ of S_2, where $1 \le x \le n_1$ and $1 \le y \le n_2$, we use the Needleman-Wunsch algorithm[20] to globally align them in a way that we create a matrix $\mathcal{N}_{x,y}$ of size $(n_1 - x + 2) \times (n_2 - y + 2)$ such that each entry $\mathcal{N}_{x,y}(i', j')$, where $1 \le i' \le n_1 - x + 1$ and $1 \le j' \le n_2 - y + 1$, represents the optimal score of globally aligning $S_1[x, x + i' - 1]$ with $S_2[y, y + j' - 1]$. Then we have $\mathcal{M}(x, i; y, j) = \mathcal{N}_{x,y}(i - x + 1, j - y + 1)$. Since the creation of $\mathcal{N}_{x,y}$ costs $\mathcal{O}((n_1 - x + 1)(n_2 - y + 1))$ time, the total time-complexity of this pre-processing stage is $\sum_{1 \le x \le n_1} \sum_{1 \le y \le n_2} \mathcal{O}((n_1 - x + 1)(n_2 - y + 1)) = \mathcal{O}(n_1^2 n_2^2) = \mathcal{O}(n^4)$, where $n = \max\{n_1, n_2\}$.

Now, we describe the details of Algorithm C2MA as follows.

Algorithm C2SA

Input: Sequences $S_1 = s_i^1 s_2^1 \cdots s_{n_1}^1$ and $S_2 = s_1^2 s_2^2 \cdots s_{n_2}^2$, and a common subsequence $P = p_1 p_2 \cdots p_\alpha$.

Output: The optimal score of the constrained sequence alignment of S_1 and S_2 with respect to P.

```
1. /* Computation of $\mathcal{M}_1(i,j)$ by Lemma 1 */
   for $i = 1$ to $n_1$ do
      for $j = 1$ to $n_2$ do
         if $s_i^1 = s_j^2 = p_1$ then
            $\mathcal{M}_1(i,j) = \mathcal{M}_0(1, i-1; 1, j-1) + f(s_i^1, s_j^2)$.
         else
            $\mathcal{M}_1(i,j) = -\infty$.
         end if
```

```
    end for
  end for
2. /* Computation of $\mathcal{M}_k(i,j)$ by Lemma 2, where $2 \le k \le \alpha$ */
  for $k = 2$ to $\alpha$ do
    for $i = 1$ to $n_1$ do
      for $j = 1$ to $n_2$ do
        if $s_i^1 = s_j^2 = p_k$ then
          $\mathcal{M}_k(i,j) = \max_{1 \le x < i, 1 \le y < j}\{\mathcal{M}_{k-1}(x,y) +$
          $\mathcal{M}_0(x+1, i-1; y+1, j-1) + f(s_i^1, s_j^2)\}$.
        else
          $\mathcal{M}_k(i,j) = -\infty$.
        end if
      end for
    end for
  end for
3. /* Computation of $\mathcal{M}_C(S_1,S_2)$ by Lemma 3 */
  $\mathcal{M}_C(S_1,S_2) = \max_{1 \le i \le n_1, 1 \le j \le n_2}\{\mathcal{M}_\alpha(i,j) + \mathcal{M}_0(i+1, n_1; j+1, n_2)\}$.
```

The correctness of Algorithm C2SA immediately follows from Lemmas 1, 2 and 3. In the following, we analyze its time-complexity. As mentioned before, the computation of $\mathcal{M}_0(x, i; y, j)$ for all $1 \le x \le i \le n_1$ and $1 \le y \le j \le n_2$ can be done in $\mathcal{O}(n_1^2 n_2^2) = \mathcal{O}(n^4)$ time, where $n = \max\{n_1, n_2\}$, in the pre-processing stage. Clearly, the cost of step 1 is $\mathcal{O}(n_1 n_2)$ time. In worst case, step 2 can be done in $\mathcal{O}(\alpha n_1^2 n_2^2)$ time. Step 3 takes $\mathcal{O}(n_1 n_2)$ time. Hence, the total time-complexity of Algorithm C2SA is $\mathcal{O}(\alpha n_1^2 n_2^2) = \mathcal{O}(\alpha n^4)$. Therefore, we have the following theorem immediately.

Theorem 1. *The constrained pairwise sequence alignment problem can be solved in $\mathcal{O}(\alpha n^4)$ time.*

4. Constrained Multiple Sequence Alignment

In this section, we design a constrained multiple sequence alignment based on the progressive approaches using Algorithm C2SA we developed in the previous section as the kernel. In general, the ideas behind the progressive strategies are as follows.[7,9,13,28,29]

1. Compute the distance matrix by aligning all pairs of sequences: Usually, this distance matrix is obtained by applying FASTA[18,21] or the dynamic programming algorithm of Needleman and Wunsch[20] to each pair of sequences.
2. Construct the guide tree from the distance matrix: For the existing progressive methods, they mainly differ in the method used to build the guide tree for directing the order of alignment of sequences. To build the guide tree, for example, PILEUP (a program of GCG packages) uses UPGMA (Unweighted

Pair-Group Method using Arithmetic mean) method[25] and CLUSTAL W[29] uses NJ (Neighbor-Joining) method.[24]

3. Progressively align the sequences according to the branching order in the guide tree: Initially, the closest two sequences in the tree are aligned using the normal dynamic programming algorithm. After aligning, this pair of sequences is fixed and any introduced gaps cannot be shifted later (i.e. once a gap, always a gap). Then the next two closest pre-aligned groups of sequences are joined in the same way until all sequences have been aligned. (Here, we may consider a sequence as an aligned group of a sequence.) To align two groups of the pre-aligned sequences, the score between any two positions in these two groups is usually the arithmetic average of the scores for all possible character comparisons at those positions. We call this kind of scoring method as a *set-to-set scoring*.

In fact, MST has been used as a significant tool for data classification in the fields of pattern recognition,[8] image processing[36] and biological data analysis.[26] As demonstrated by Xu, Olman and Xu,[35] they discovered a new relationship between MST and clustering, and had developed an MST-based clustering package called EXCAVATOR for successfully and efficiently clustering gene expression data. In this section, we propose a variant of progressive method by using the Kruskal MST to construct the guide tree, called Kruskal merging order tree, and using our developed Algorithm C2SA to join the pre-aligned groups of sequences. We call this kind of progressive tool as CMSA. The Kruskal merging order tree of K sequences is constructed as follows. First, we create a complete graph $G = (V, E)$ of K sequences in a way that each vertex of V represents a sequence and each edge e of E is associated with a weight $d(e)$ to represent the distance between the corresponding sequences of its end-vertices. Then we use the Kruskal's algorithm[15] to construct the Kruskal MST of G, denoted by $\mathcal{T}$. For completeness, we describe the Kruskal method for constructing $\mathcal{T}$ as follows.

1. Sort all edges of E in non-decreasing order according to their distances.
2. Initially, $\mathcal{T}$ is empty. Then we repeatedly add the edges of E in non-decreasing order to $\mathcal{T}$ in a way that if the currently adding edge e to $\mathcal{T}$ does not create a cycle in $\mathcal{T}$, then we add e to $\mathcal{T}$; otherwise, we discard e.

Next, according to the Kruskal MST $\mathcal{T}$, we build the Kruskal merging order tree $\mathcal{T}_K$ as follows.

1. Let $V = \{v_1, v_2, \ldots, v_K\}$ and $e_1, e_2, \ldots, e_{K-1}$ be the edges of T with $d(e_1) \leq d(e_2) \leq \cdots \leq d(e_{K-1})$.
2. For each vertex $v_i \in V$, we create a tree $\mathcal{T}_i$ such that $\mathcal{T}_i$ contains only a node v_i. For the purpose of merging trees, we consider $\mathcal{T}_i$ as a *rooted tree* by designating v_i as its root, and define the *merge* of two tree $\mathcal{T}_i$ and $\mathcal{T}_j$ respectively rooted at v_i and v_j to be a new tree rooted at a new vertex u such that v_i and v_j become the children of u.

3. For each $e_k = (v_i, v_j)$, where k increases from 1 to $K-1$, we find the trees $\mathcal{T}_i$ and $\mathcal{T}_j$ containing v_i and v_j respectively and then merge them into a new tree. This process is continued until the remaining is only one tree. Then this final tree is the so-called *Kruskal merging order tree* $\mathcal{T}_K$.

Clearly, the construction of G for K sequences can be done in $\mathcal{O}(K^2)$ time and the computation of the Kruskal's MST $\mathcal{T}$ of G can be done in $\mathcal{O}(K^2 \log K)$ time.[15] Then the construction of $\mathcal{T}_K$ from $\mathcal{T}$ can be implemented by the disjoint set union and find algorithm proposed by Gabow and Tarjan[10] in $\mathcal{O}(m+K)$ time, where m denotes the number of union and find operations. It is not hard to see that $m = \mathcal{O}(K)$ and hence the construction of $\mathcal{T}_K$ takes $\mathcal{O}(K)$ time. Therefore, the total time-complexity of constructing the Kruskal merging order tree $\mathcal{T}_K$ from K sequences is $\mathcal{O}(K^2 \log K)$.

Now, we describe the detailed algorithm of our CMSA in the following.

Algorithm CMSA

Input: K sequences $S_1, S_2, \ldots, S_K$ and a common subsequence $P = p_1 p_2 \cdots p_\alpha$.

Output: The constrained multiple sequence alignment of $S_1, S_2, \ldots, S_K$ with respect to P.

(1) Compute the distance matrix D by globally aligning all pairs of sequences without any constraint using the Needleman-Wunsch algorithm such that $D(i,j)$ denotes the distance between sequences S_i and S_j.

(2) Create a complete graph G from the distance matrix D and then compute the Kruskal merging order tree $\mathcal{T}_K$ from G to serve as the guide tree.

(3) Progressively align the sequences according to the branching order of the guide tree $\mathcal{T}_K$ in a way that the currently two closest pre-aligned groups of sequences are joined by applying Algorithm C2SA to the represented sequences of these two groups, where we arbitrarily choose one sequence in each group as the represented sequence in which we further classify each character s, which equals to a constrained character of P, in the represented sequence as a *real-constraint* if all characters aligned with s in the group are the same; otherwise, as a *pseudo-constraint*. Then we use the score between two represented sequences to denote that between two pre-aligned groups of sequences. Hence, we call this kind of scoring method as *peer-to-peer scoring*.

Actually, our merging method in step 3 of Algorithm CMSA is similar to a heuristic approach used to solve the so-called tree alignment, which is known to be NP-complete,[32] but a lot of heuristic algorithms[1,12,23] and even PTASs[31,33,34] have been discussed in the literature. In the following, we analyze the time-complexity of Algorithm CMSA. Clearly, step 1 can be done in $\mathcal{O}(K^2 n^2)$, where $n = \max\{n_i : 1 \le i \le K\}$. As mentioned before, step 2 costs $\mathcal{O}(K^2 \log K)$ time. In step 3, there are at most $\mathcal{O}(K)$ iterations for calling Algorithm C2SA, whose time-complexity is $\mathcal{O}(\alpha n^4)$ by Theorem 1, to join two pre-aligned groups of sequences. Hence, the time-complexity of step 3 is $\mathcal{O}(\alpha K n^4)$. Clearly, the cost of Algorithm CMSA is dominated by step 3 and hence its time-complexity is $\mathcal{O}(\alpha K n^4)$.

Theorem 2. *Given K sequences $S_1, S_2, \ldots, S_K$ and a common subsequence P of $S_1, S_2, \ldots, S_K$ with length α, the time-complexity of Algorithm* CMSA *is $\mathcal{O}(\alpha K n^4)$.*

5. Experimental Results

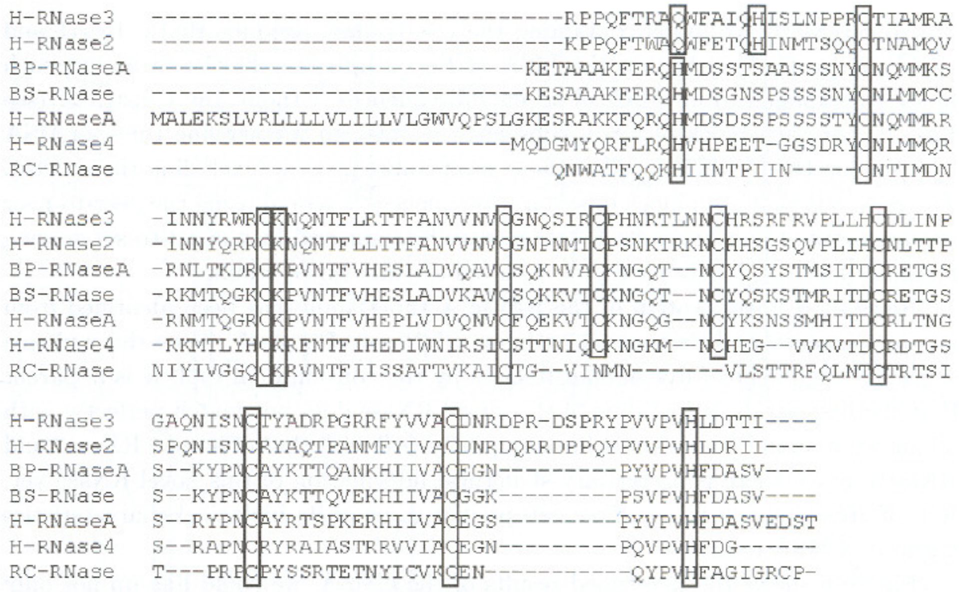
In this section, we describe the motivation of developing the CMSA tool in more details and report the achievement we have accomplished by CMSA tool we developed to analyze seven RNases with known structures.

RNase (ribonuclease) catalyzes the degradation of RNAs (ribonucleic acids) including ribosomal RNA (rRNA), messenger RNA (mRNA), and transfer RNA (tRNA). All living organisms contain RNases which mainly function in the RNA processing such as rRNA maturation, tRNA maturation, and mRNA maturation and turnover. The amino acid sequences and the correspondent gene sequences for many RNases have been reported in a couple of databases. For example, one can obtain 225,378, and 1,925 various entries of RNases from protein data bank (PDB), genebank, and National Center for Biotechnology Information (NCBI), respectively. Three dimensional structural determinations by X-ray and NMR have revealed that bovine pancreatic RNase A (BP-RNaseA) is composed of three α helices, 14 β sheets with no turn and loop. The other RNases also possess similar spatial arrangement as shown in 47 solved structures among which eight representative structures could be obtained. Most RNases act as a monomer with only a few exceptions whose dimer forms show biological function.

Of the overall several hundreds of RNases identified and characterized, it has been found that nine of the RNases also possess toxicity. The others are non-toxic proteins. The toxic RNases come from different origins including human RNase2, RNase3, and RNase4, bovine seminal RNase (BS-RNase), bull frog onconase, RC-RNase, and RJ-RNase, Aspigillus a-sarcin, and bacterial barnase. Some of the toxic RNases show normal RNase activities and some possess much lower RNase activities. As for the toxicity, it has been reported that these RNases show cytotoxicity, neurotoxicity, angiogenesis, or antisarcoma activities. Interestingly, some toxic RNases require RNase activity for their cytotoxicity, but the RNase activity is not essential for some others. Nevertheless, the three-dimensional structures of these RNases showed very similar pattern.

Since all RNases appear to be the products of molecular evolution, when and how other biological function incorporates into the molecule remain to be answered. It is of great interest to investigate the structure-function relationship among all RNases with well defined 3D structures. In fact, when all the available RNase sequences were analyzed, no obvious homology could be obtained. We thus chose seven RNases, including the non-toxic human and bovine pancreatic RNaseAs and toxic human RNase2, RNase3, RNase4, BS-RNase and bullfrog RC-RNase, to compare their primary sequences. As compared to bovine pancreatic RNaseA, the major structural features of all these RNases contain three conserved active site residues (His12, Lys41 and His119) and four disulfide bonds. However, their RNase activities and substrate specificity vary from species to species. Since the solved 3D structures of these proteins all show very high homology among the catalytic domains and disulfide linkages, we would expect that their primary sequence comparisons to be matched very well. Initially we compared these sequences pairwisely with the conventional "Workbench 3.2" tool (http://workbench.sdsc.edu/), all the sequences lined up well with the BP-RNaseA. However, when all seven sequences were compared using the same tool, some mismatches occurred as shown in Fig. 1. It seemed that all the sequences lined up very well, as the conserved cysteines (Cys), Lys41 and His119 residues could be identified. However, the key catalytic residue His12 of human RNase2 and RNase3 lined up with different amino acids in the other five sequences. Obviously this result was not satisfactory since His12 of all RNases has been biologically proved to be an important active site residue and should thus be well conserved among all RNases.

Figure 2 shows the result we obtained by aligning all above seven sequences using our CMSA tool, where we let P = HKH. As shown in the figure, all residues correspondent to the His12, Lys41, His119 of BP-RNaseA were lined up very well, and so were all the Cys residues.



```
H-RNase3   -------------------------------RPPQFTRAQWFAIQHISLNPPRCTIAMRA
H-RNase2   -------------------------------KPPQFTWAQWFETQHINMTSQQCTNAMQV
BP-RNaseA  ----------------------------KETAAAKFERQHMDSSTSAASSSNYCNQMMKS
BS-RNase   ----------------------------KESAAAKFERQHMDSGNSPSSSSNYCNLMMCC
H-RNaseA   MALEKSLVRLLLLVLILLVLGWVQPSLGKESRAKKFQRQHMDSDSSPSSSSTYCNQMMRR
H-RNase4   ---------------------------MQDGMYQRFLRQHVHPEET-GGSDRYCNLMMQR
RC-RNase   ------------------------------QNWATFQQKHIINTPIIN-----CNTIMDN

H-RNase3   -INNYRWRCKNQNTFLRTTFANVVNVCGNQSIRCPHNRTLNNCHRSRFRVPLLHCDLINP
H-RNase2   -INNYQRRCKNQNTFLLTTFANVVNVCGNPNMTCPSNKTRKNCHHSGSQVPLIHCNLTTP
BP-RNaseA  -RNLTKDRCKPVNTFVHESLADVQAVCSQKNVACKNGQT--NCYQSYSTMSITDCRETGS
BS-RNase   -RKMTQGKCKPVNTFVHESLADVKAVCSQKKVTCKNGQT--NCYQSKSTMRITDCRETGS
H-RNaseA   -RNMTQGRCKPVNTFVHEPLVDVQNVCFQEKVTCKNGQG--NCYKSNSSMHITDCRLTNG
H-RNase4   -RKMTLYHCKRFNTFIHEDIWNIRSICSTTNIQCKNGKM--NCHEG--VVKVTDCRDTGS
RC-RNase   NIYIVGGQCKRVNTFIISSATTVKAICTG--VINMN-------VLSTTRFQLNTCTRTSI

H-RNase3   GAQNISNCTYADRPGRRFYVVACDNRDPR-DSPRYPVVPVHLDTTI----
H-RNase2   SPQNISNCRYAQTPANMFYIVACDNRDQRRDPPQYPVVPVHLDRII----
BP-RNaseA  S--KYPNCAYKTTQANKHIIVACEGN---------PYVPVHFDASV----
BS-RNase   S--KYPNCAYKTTQVEKHIIVACGGK---------PSVPVHFDASV----
H-RNaseA   S--RYPNCAYRTSPKERHIIVACEGS---------PYVPVHFDASVEDST
H-RNase4   S--RAPNCRYRAIASTRRVVIACEGN---------PQVPVHFDG------
RC-RNase   T---PRPCPYSSRTETNYICVKCEN-----------QYPVHFAGIGRCP-
```

Fig. 1. The multiple sequence alignment of seven RNases by WorkBench 3.2: The key active site residues homologous to His12, Lys41, and His119 of BP-RNaseA, the cysteine residues responsible for disulfide bond linkage and two matched Gln residues are shown in boxes.

```
H-RNase3    -RP--PQFTRAQWFAIQHIS-L-NPP---R--CTIAMRAI---NN--Y--RWRCKNQNTF
H-RNase2    MKP--PQFTWAQWFETQHIN-M-TSQ---Q--CTNAMQVI---NN--Y-QR-RCKNQNTF
BP-RNaseA   -KETAA----AK-FERQHMD-SSTSAASSSNYC-N--QMMKSRN---LTKD-RCKPVNTF
BS-RNase    -KESAA----AK-FERQHMD-SGNSPSSSSNYC-N--LMMCCRK---MTQG-KCKPVNTF
H-RNaseA    -KES-R----AKAFQRQHMD-SDSSPSSSSTYC-N--QMM-RRRN--MTQG-RCKPVNTF
H-RNase4    -MQDGMY---QR-FLRQHVHPEET--GGSDRYC-N--LMMQRRK---MTLY-HCKRFNTF
RC-RNase    --XN-W----A-TFQQKHI--I-NT-PIIN--C-N--TIM--DNNIYIVGG-QCKRVNTF

H-RNase3    LRTTFANVVNVCGNQSIRCPHNRTLNNCHRSRFRVPL-LHC-DLINP-GAQNISNCRYAD
H-RNase2    LLTTFANVVNVCGNPNMTCPSNKTRKNCHHSGSQVPL-IHC-NLTTP-SPQNISNCRYAQ
BP-RNaseA   VHESLADVQAVCSQKNVACK-N-GQTNCYQSYSTMSI-TDC-RET-GSSKYP--NCAY-K
BS-RNase    VHESLADVKAVCSQKKVTCK-N-GQTNCYQSKSTMRI-TDC-RET-GSSKYP--NCAY-K
H-RNaseA    VHEPLVDVQNVCFQEKVTCK-N-GQGNCYKSNSSMHI-TDC-RLTNG-SRYP--NCAY-R
H-RNase4    IHEDIWNIRSICSTTNIQCK-N-GKMNCHE--GVVKV-TDC-RDT-GSSRAP--NCRY-R
RC-RNase    IISSATTVKAIC--TGV-I--N--M-NVL-STTRFQLNT-CTR-TSI-TP-R--PCPY--

H-RNase3    R-PGR-RFYVVACDNRD-PRDSPR-YPVVPVHLDTTI----
H-RNase2    T-PAN-MFYIVACDNRDQRRD-PPQYPVVPVHLD-RI----
BP-RnaseA   TTQAN-KHIIVACEG-----N---PY--VPVHFDASV----
BS-RNase    TTQVE-KHIIVACGG-----K---PS--VPVHFDASV----
H-RNaseA    TSPKE-RHIIVACEG-----S---PY--VPVHFDASV----
H-RNase4    AI-ASTRRVVIACEG-----N---PQ--VPVHFD-G-----
RC-RNase    SSRTETNYICVKCE------N---QY---PVHF-AGIGRCP
```

Fig. 2. The multiple sequence alignment of seven RNases by our CMSA: The key active site residues homologous to His12, Lys41, and His119 of BP-RNaseA, the cysteine residues responsible for disulfide bond linkage, and two matched Gln residues are shown in boxes.

Note that the columns containing the constrained residues His12, Lys41 and His119 respectively partition the MSA of Fig. 2 into four blocks such that each block corresponds to an MSA of seven subsequences without any constraint (see Fig. 3). For each block of seven subsequences, we can further fine tune its MSA by applying the CLUSTAL W to these seven subsequences (excluding the spaces). The reason is that each block of seven subsequences is scored using the peer-to-peer scoring method and hence its SP-score may be improved by the set-to-set scoring method, like CLUSTAL W. Figure 4 shows the fine-tuned result.

Recently a novel protein of the RNase A superfamily has been identified from human placenta and designated as human RNase8. Figure 5 shows the multiple sequence alignment of five human RNases by the conventional tool. It is apparent that the first catalytic histidine of this novel RNase does not match perfectly with all known human RNase, but it lines up pretty well with that of human RNase2 and RNase3. Since there is no tertiary structural information of this novel RNase yet, it is of great interest to see if we can predict some of its biological characteristics with our CMSA tool.

Figure 6 shows the fine-tuned results of the CMSA, we could line up not only the cysteine residues but also the catalytic residues His12, Lys41, and His119. We have demonstrated that although the overall sequence homology of this novel human RNase8 is quite low as compared with other members of the human RNase family, its

```
                   Block 1                          Block 2                      Block 3
H-RNase3   -RP--PQFTRAQWFAIQHIS-L-NPP---R--CTIAMRAI---NN--Y--RWRCKNQNTF
H-RNase2   MKP--PQFTWAQWFETQHIN-M-TSQ---Q--CTNAMQVI---NN--Y-QR-RCKNQNTF
BP-RNaseA  -KETAA----AK-FERQHMD-SSTSAASSSNYC-N--QMMKSRN---LTKD-RCKPVNTF
BS-RNase   -KESAA----AK-FERQHMD-SGNSPSSSSNYC-N--LMMCCRK---MTQG-KCKPVNTF
H-RNaseA   -KES-R----AKAFQRQHMD-SDSSPSSSSTYC-N--QMM-RRRN--MTQG-RCKPVNTF
H-RNase4   -MQDGMY---QR-FLRQHVHPEET--GGSDRYC-N--LMMQRRK---MTLY-HCKRFNTF
RC-RNase   --XN-W----A-TFQQKHI--I-NT-PIIN--C-N--TIM--DNNIYIVGG-QCKRVNTF

H-RNase3   LRTTFANVVNVCGNQSIRCPHNRTLNNCHRSRFRVPL-LHC-DLINP-GAQNISNCRYAD
H-RNase2   LLTTFANVVNVCGNPNMTCPSNKTRKNCHHSGSQVPL-IHC-NLTTP-SPQNISNCRYAQ
BP-RNaseA  VHESLADVQAVCSQKNVACK-N-GQTNCYQSYSTMSI-TDC-RET-GSSKYP--NCAY-K
BS-RNase   VHESLADVKAVCSQKKVTCK-N-GQTNCYQSKSTMRI-TDC-RET-GSSKYP--NCAY-K
H-RNaseA   VHEPLVDVQNVCFQEKVTCK-N-GQGNCYKSNSSMHI-TDC-RLTNG-SRYP--NCAY-R
H-RNase4   IHEDIWNIRSICSTTNIQCK-N-GKMNCHE--GVVKV-TDC-RDT-GSSRAP--NCRY-R
RC-RNase   IISSATTVKAIC--TGV-I--N--M-NVL-STTRFQLNT-CTR-TSI-TP-R--PCPY--

H-RNase3   R-PGR-RFYVVACDNRD-PRDSPR-YPVVPVHLDTTI
H-RNase2   T-PAN-MFYIVACDNRDQRRD-PPQYPVVPVHLD-RI----
BP-RnaseA  TTQAN-KHIIVACEG-----N---PY--VPVHFDASV----
BS-RNase   TTQVE-KHIIVACGG-----K---PS--VPVHFDASV----
H-RNaseA   TSPKE-RHIIVACEG-----S---PY--VPVHFDASV----
H-RNase4   AI-ASTRRVVIACEG-----N---PQ--VPVHFD-G-----
RC-RNase   SSRTETNYICVKCE------N---QY---PVHF-AGIGRCP
                     Block 3                   Block 4
```

Fig. 3. Three columns containing His12, Lys41 and His119 respectively partition the MSA of Fig. 2 into four blocks such that each block corresponds to an MSA of seven subsequences without any constraint.

```
H-RNase3   -RPPQFTRAQWFAIQH------ISLNPPRCTIAMRA-INNYRWRCKNQNTFLRTTFANV
H-RNase2   MKPPQFTWAQWFETQH------INMTSQQCTNAMQV-INNYQRRCKNQNTFLLTTFANV
BP-RNaseA  ----KETAAAKFERQHMDSSTSAASSSNYCNQMMKS-RNLTKDRCKPVNTFVHESLADV
BS-RNase   ----KESAAAKFERQHMDSGNSPSSSSNYCNLMMCC-RKMTQGKCKPVNTFVHESLADV
H-RNaseA   ----KESRAKAFQRQHMDSDSSPSSSSTYCNQMMRR-RNMTQGRCKPVNTFVHEPLVDV
H-RNase4   ---MQDGMYQRFLRQH-VHPEETGGSDRYCNLMMQR-RKMTLYHCKRFNTFIHEDIWNI
RC-RNase   ------XNWATFQQKH-----IINTPIINCNTIMDNNIYIVGGQCKRVNTFIISSATTV

H-RNase3   VNVCGNQSIRCPHNRTLNNCHRSRFRVPLLHCDLINPGAQNISNCRYADRPGRRFYVVA
H-RNase2   VNVCGNPNMTCPSNKTRKNCHHSGSQVPLIHCNLTTPSPQNISNCRYAQTPANMFYIVA
BP-RNaseA  QAVCSQKNVACKNGQT--NCYQSYSTMSITDCRETGSS--KYPNCAYKTTQANKHIIVA
BS-RNase   KAVCSQKKVTCKNGQT--NCYQSKSTMRITDCRETGSS--KYPNCAYKTTQVEKHIIVA
H-RNaseA   QNVCFQEKVTCKNGQG--NCYKSNSSMHITDCRLTNGS--RYPNCAYRTSPKERHIIVA
H-RNase4   RSICSTTNIQCKNGKM--NCHEG--VVKVTDCRDTGSS--RAPNCRYRAIASTRRVVIA
RC-RNase   KAICTGVINMN---------VLSTTRFQLNTCTRTSITP---RPCPYSSRTETNYICVK

H-RNase3   CDNRDPR-DSPRYPVVPVHLDTTI---
H-RNase2   CDNRDQRRDPPQYPVVPVHLDRI----
BP-RNaseA  CEG---------NPYVPVHFDASV---
BS-RNase   CGG---------KPSVPVHFDASV---
H-RNaseA   CEG---------SPYVPVHFDASV---
H-RNase4   CEG---------NPQVPVHFDG-----
RC-RNase   CEN-----------QYPVHFAGIGRCP
```

Fig. 4. The multiple sequence alignment of seven RNases by our CMSA which is further fine tuned by applying the CLUSTAL W to each block of seven subsequences excluding the spaces in Fig. 3.

```
H-RNase4  ---------------------------MQDGMYQRFLRQHVHPEET-GGSDRYCNLMMQR
H-RNaseA  MALEKSLVRLLLLVLILLVLGWVQPSLGKESRAKKFQRQHMDSDSSPSSSSTYCNQMMRR
H-RNase3  -------------------------------RPPQFTRAQWFAIQHISLNPPRCTIAMRA
H-RNase2  -------------------------------KPPQFTWAQWFETQHINMTSQQCTNAMQV
H-RNase8  ----MAPARAGCCPLLLLLLGLWVAEVLVRAKPKDMTSSQWFKTQHVQPSPQACNSAMSI

H-RNase4  RKMTLYHCKRFNTFIHEDIWNIRSICSTTNIQCKN--GKMNCHEG--VVKVTDCRDTGSS
H-RNaseA  RNMTQGRCKPVNTFVHEPLVDVQNVCFQEKVTCKN--GQGNCYKSNSSMHITDCRLTNGS
H-RNase3  INNYRWRCKNQNTFLRTTFANVVNVCGNQSIRCPHNRTLNNCHRSRFRVPLLHCDLINPG
H-RNase2  INNYQRRCKNQNTFLLTTFANVVNVCGNPNMTCPSNKTRKNCHHSGSQVPLIHCNLTTPS
H-RNase8  INKYTERCKDLNTFLHEPFSSVAITCQTPNIACKN--SCKNCHQSHGPMSLTMGELTSG-

H-RNase4  --RAPNCRYRAIASTRRVVIACE---------GNPQVPVHFDG------
H-RNaseA  --RYPNCAYRTSPKERHIIVACE---------GSPYVPVHFDASVEDST
H-RNase3  AQNISNCTYADRPGRRFYVVACDNRDPR-DSPRYPVVPVHLDTTI----
H-RNase2  PQNISNCRYAQTPANMFYIVACDNRDQRRDPPQYPVVPVHLDRII----
H-RNase8  --KYPNCRYKEKHLNTPYIVACDPPQQ--GDPGYPLVPVHLDKVV----
```

Fig. 5. The multiple sequence alignment of five human RNases by the conventional tool.

```
H-RNase4  -----------------------------MQDGMYQRFLRQH-VHPEETGGSDRYCN
H-RNaseA  ------------------------------KESRAKAFQRQHMDSDSSPSSSSTYCN
H-RNase3  ---------------------------RPPQFTRAQWFAIQH------ISLNPPRCT
H-RNase2  --------------------------MKPPQFTWAQWFETQH------INMTSQQCT
H-RNase8  MAPARAGCCPLLLLLLGLWVAEVLVRAKPKDMTSSQWFKTQH------VQPSPQACN

H-RNase4  LMMQRRKMTLYHCKRFNTFIHEDIWNIRSICSTTNIQCKNG--KMNCHEG--VVKVT
H-RNaseA  QMMRRRNMTQGRCKPVNTFVHEPLVDVQNVCFQEKVTCKNG--QGNCYKSNSSMHIT
H-RNase3  IAMRAINNYRWRCKNQNTFLRTTFANVVNVCGNQSIRCPHNRTLNNCHRSRFRVPLL
H-RNase2  NAMQVINNYQRRCKNQNTFLLTTFANVVNVCGNPNMTCPSNKTRKNCHHSGSQVPLI
H-RNase8  SAMSIINKYTERCKDLNTFLHEPFSSVAITCQTPNIACKNS--CKNCHQSHGPMSLT

H-RNase4  DCRDTGSS--RAPNCRYRAIASTRRVVIACE---------GNPQVPVHFDG------
H-RNaseA  DCRLTNGS--RYPNCAYRTSPKERHIIVACE---------GSPYVPVHFDASVEDST
H-RNase3  HCDLINPGAQNISNCRYADRPGRRFYVVACDNRDPR-DSPRYPVVPVHLDTTI----
H-RNase2  HCNLTTPSPQNISNCRYAQTPANMFYIVACDNRDQRRDPPQYPVVPVHLDRI-----
H-RNase8  MGELTSG---KYPNCRYKEKHLNTPYIVACDP--PQQGDPGYPLVPVHLDKVV----
```

Fig. 6. The fine-tuned multiple sequence alignment of five human RNases by our CMSA.

biological function is possibly most similar to that of human RNase2 and RNase3. In other words, this novel protein may also possess some biological functions in addition to its normal RNase activity.

Since the accumulation of the data is much faster by computer than by biological experiments, the customerized CMSA RNase tool can assist the biologists to quickly predict the biological function of novel proteins in this superfamily. The rapid evolution of the RNase family can be thus studied with a novel and convenient tool.

6. Conclusions

In this paper, we designed a CMSA tool for guaranteeing that the generated alignment satisfies the user-specified constraints that some particular residues should be aligned together. The time-complexity of our CMSA algorithm for aligning K sequences is $\mathcal{O}(\alpha K n^4)$, where n is the maximum of the lengths of sequences and α is the number of residues needed to be aligned together. We also experimented our developed CMSA tool on the RNases sequences with known structures and the results illustrated the practicability of our method since all the important active site residues were well aligned. We believe that the development of such a CMSA tool may lead the biologists to integrate more structure constraints to generate a novel database of RNases, and hence fast and precise predictions of the biological functions of most RNases can be achieved.

Enzymes are proteins capable of carrying out complex biochemical transformations in aqueous solution at biological temperatures and pH in a stereospecific manner. It is known that only the enzymes with correct folding or tertiary structural conformation can catalyze biochemical transformation. Since the special arrangement of a particular tertiary structure is determined by specific interactions among the primary sequences and among the secondary structures, the high homology among the 3D structures of an enzyme family should reflect even higher order of similarity among their secondary or primary structures. The biologists usually just use the default parameters to make sequence alignment employing the conventional analytical tools. To incorporate more structural and functional information to the multiple sequence alignment, it is important to find out the specific location in the primary sequence of an enzyme and add more knowledge-based constraints in the algorithm. The key catalytic residues of enzymes are usually well conserved in a superfamily. For example, the Ser-Asp-His catalytic triad residues of the serine proteases,[30] the Cys-His catalytic diad residues of the cysteine proteinases,[14] the Arg-Asp-His catalytic triad residues of the phospholipases,[16] and the Ser-His-Asp catalytic triad of the carboxypeptidases[3] have all been identified to be well conserved as in the case of our RNase superfamily which possess the conserved His-Lys-His catalytic residues. Therefore, we can select the constrained sequences based on the key catalytic residues of the enzyme superfamily of interest. Since the three-dimensional structures of the RNase family show high conservation of the

His-Lys-His catalytic residues in space, the alignment of the primary sequences of these enzymes by our CMSA indeed reflect the homology. The specific example in this practice showed that the major characteristics of an RNase could be found only when CMSA was applied to perform multiple sequence alignment. Based on the new comparison results, the biologists can further investigate the evolutionary relationship of the RNase family and the essential residues required for other function of the RNases.

In fact, each of the columns requested to be aligned well in our CMSA model can represent a conserved site of a protein family. To our knowledge, each conserved site may consist of a short segment of amino acids, instead of a single amino acid, and often a conserved site does not occur exactly in each protein of the family. This kind of CMSA model is more complex than that we discussed in this paper and it would be of interest to design a new package to tackle this model.

In the following, we propose two open problems concerning our progressive method for the future works.

(1) Whether our heuristic method can become an approximation algorithm of a constant performance ratio?
(2) Whether the time-complexity of our algorithm can be further improved?

From the practical viewpoint, we may be able to speedup our programs by using the special data structures like hashing tables, or using the dedicated hardware of special purpose like PARACEL of Celera Genomics (http://www.paracel.com/).

Acknowledgments

The authors would like to thank the anonymous referees for many constructive suggestions for the presentation of this paper.

References

1. S. F. Altschul and D. J. Lipman, "Trees, stars and multiple biological sequence alignment," *SIAM J. Appl. Math.* **49**, 197–209 (1989).
2. V. Bafna, E. L. Lawler and P. A. Pevzner, "Approximation algorithms for multiple sequence alignment," *Theo. Comp. Sci.* **182**, 233–244 (1997).
3. G. J. Bartlett, C. T. Porter, N. Borkakoti and J. M. Thornton, "Analysis of catalytic residues in enzyme active sites," *J. Mol. Biol.* **324**, 105–121 (2002).
4. P. Bonizzoni and G. D. Vedova, "The complexity of multiple sequence alignment with SP-score that is a metric," *Theo. Comp. Sci.* **259**, 63–79 (2001).
5. H. Carrillo and D. Lipman, "The multiple sequence alignment problem in biology," *SIAM J. Appl. Math.* **48**, 1073–1082 (1988).
6. S. C. Chan, A. K. C. Wong and D. K. Y. Chiu, "A survey of multiple sequence comparison methods," *B. Math. Biol.* **54**, 563–598 (1992).
7. F. Corpet, "Multiple sequence alignment with hierarchical clustering," *Nucleic Acids Res.* **16**, 10881–10890 (1988).
8. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley-Interscience, New York, 1973).

9. D. F. Feng and R. F. Doolittle, "Progressive sequence alignment as a prerequisite to correct phylogenetic trees," *J. Mol. Evol.* **25**, 351–360 (1987).
10. H. N. Gabow and R. E. Tarjan, "A linear-time algorithm for a special case of disjoint set union," *J. Comput. Syst. Sci.* **30**, 209–221 (1985).
11. D. Gusfield, "Efficient methods for multiple sequence alignment with guaranteed error bounds," *B. Math. Biol.* **55**, 141–154 (1993).
12. J. Hein, "A new method that simultaneously aligns and reconstruct ancestral sequences for any number of homologous sequences, when the phylogeny is given," *Mol. Biol. Evol.* **6**, 649–668 (1989).
13. D. Higgins and P. Sharpe, "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer," *Gene* **73**, 237–244 (1988).
14. J. Kronovetr and T. Skern, "Foot-and-mouth disease virus leader proteinase: a papain-like enzyme requiring an acidic environment in the active site," *FEBS Letters* **528**, 58–62 (2002).
15. J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Am. Math. Soc.* **7**, 48–50 (1956).
16. R. J. Kubiak, X. Yue, R. J. Hondal, C. Mihai, M. D. Tsai and K. S. Bruzik, "Involvement of the Arg-Asp-His catalytic triad in enzymatic cleavage of the phosphodiester bond," *Biochemistry* **40**, 5422–5432 (2001).
17. M. Li, B. Ma and L. Wang, "Near optimal multiple alignment within a band in polynomial time," In *Proceedings of the Thirty Second Annual ACM Symposium on Theory of Computing* (*STOC 2000*), pp. 425–434. ACM Press, Portland, 2000.
18. D. J. Lipman and W. R. Pearson, "Rapid and sensitive protein similarity search," *Science* **227**, 1435–1441 (1985).
19. G. Myers, S. Selznick, Z. Zhang and W. Miller, "Progressive multiple alignment with constraints," *J. Comput. Biol.* **3**, 563–572 (1996).
20. S. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Evol.* **48**, 443–453 (1970).
21. W. R. Pearson, "Searching protein sequence libraries: Computation of the sensitivity and selectivity of the Smith–Waterman and FASTA algorithms," *Genomics* **11**, 635–650 (1991).
22. P. A. Pevzner, "Multiple alignment, communication cost, and graph matching," *SIAM J. Appl. Math.* **52**, 1763–1779 (1992).
23. R. Ravi and J. Kececioglu, "Approximation algorithms for multiple sequence alignment under a fixed evolutionary tree," *Discrete Appl. Math.* **88**, 355–366 (1998).
24. N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Mol. Biol. Evol.* **4**, 406–425 (1987).
25. P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy* (Freeman, San Francisco, CA, 1973).
26. D. J. States, N. L. Harris and L. Hunter, "Computationally efficient cluster representation in molecular sequence megaclassification," In L. Hunter, D. Searls and J. Shavlik, eds. *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology* (*ISMB 1993*), pp. 387–394. AAAI Press, 1993.
27. C. Y. Tang, C. L. Lu, M. D. T. Chang, Y. J. Sun, Y. T. Tsai, J. M. Chang, Y. H. Chiou, C. M. Wu, H. T. Chang, W. I. Chou and S. C. Chiang, "Constrained sequence alignment tool development and its application to rnase family alignment," In F. Valafar, ed. *Proceedings of International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences* (*METMBS 2002*), pp. 365–370, 2002.

28. W. R. Taylor, "Multiple sequence alignment by a pairwise algorithm," *CABIOS* **3**, 81–87 (1987).
29. J. D. Thompson, D. G. Higgs, and T. J. Gibson, "CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties, and weight matrix choice," *Nucleic Acids Res.* **22**, 4673–4680 (1994).
30. S. I. Tyukhtenko, A. V. Litvinchuk, C. F. Chang, R. J. Leu, J. F. Shaw and T. H. Huang, "NMR studies of the hydrogen bonds involving the catalytic triad of Escherichia coli thioesterase/protease I," *FEBS Letters* **528**, 203–206 (2002).
31. L. Wang and D. Gusfield, "Improved approximation algorithms for tree alignment," *J. Algorithm.* **25**, 255–273 (1997).
32. L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," *Journal of Computational Biology* **1**, 337–348 (1994).
33. L. Wang, T. Jiang and D. Gusfield, "A more efficient approximation scheme for tree alignment," *SIAM J.* **30**, 283–299 (2000).
34. L. Wang, T. Jiang and E. L. Lawler, "Approximation algorithms for tree alignment with a given phylogeny," *Algorithmica* **16**, 302–315 (1996).
35. Y. Xu, V. Olman and D. Xu, "Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees," *Bioinformatics* **18**, 536–545 (2002).
36. Y. Xu and E. C. Uberbacher, "2D image segmentation using minimum spanning trees," *Image Vision Comput.* **15**, 47–57 (1997).

Chuan-Yi Tang received the B.S. and M.S. degrees from National Hsing Hua University, Hsinchu, Taiwan, in 1980 and 1982, respectively, and the Ph.D. degree in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1985.

Since August 1985, he has been with the National Tsing Hua University, where he is currently a Professor and chairman at the Department of Computer Science. His research interests include analysis and design of algorithms, computational molecular biology, parallel processing and computer aided engineering. In 2000 and 2001, he obtained the award of outstanding research of NSC.

Chin Lung Lu received the B.S. degree in Information and Computer Education from National Taiwan Normal University in 1991, the M.S. degree in Computer Science and Information Engineering from National Chung Cheng University in 1993, and the Ph.D. degree in Department of Computer Science from National Tsing Hua University in 1998. Recently, he is an assistant professor in the Institute of Bioinformatics & Department of Biological Science and Technology of National Chiao Tung University. His research interests include design and analysis of algorithms, graph algorithms and computational molecular biology.

Margaret Dah-Tsyr Chang received the B.S. degree in chemistry from National Taiwan University at Taipei, Taiwan in 1987 and the Ph.D. degree in chemistry from the Johns Hopkins University at Baltimore, Maryland, USA in 1993.

She is currently an associate professor in the Institute of Molecular and Cellular Biology & Department of Life Science, National Tsing Hua University at Hsinchu, Taiwan. Her current research interests include engineering, expression, and purification of recombinant proteins, protein-protein interaction, protein structure and function analyses.

Yin-Te Tsai is currently a associate professor of Department of Computer Science and Information Management at Providence University, ShaLu, Taiwan, R.O.C. He received his B.S. degree in computer engineering from National Chiao Tung University, M.S. degree in Computer Science and information engineering from National Chiao Tung University, and Ph.D. degree in computer science from National Hsin Hua University, in 1987, 1989 and 1994, respectively. From July 1994 to May 1996, he was a computer instructor at Army Electronics and Communication School for the military service. He joined the Department of Computer Science and Information Management at Providence University in June 1996. His current research interests include the design and analysis of algorithms, bioinformatics and information systems.

Yuh-Ju Sun was born in Kau-shoun, Taiwan, in 1948. She received the B.S. degree in Chemistry from Chung-Yuan Christian University, Taiwan, R.O.C., in 1982, M.S. degree in Chemistry from Auburn University, Auburn, AL, USA, in 1987 and Ph.D. degree in crystallography, from University of Pittsburgh, Pittsburgh, PA, USA, in 1995. She is currently an assistant professor of the Department of Life Sciences, Institute of Bioinformatics and Structural Biology, National Tsing Hua University. From 1995 to 1999, she worked as a postdoctoral fellow at the Institute of Molecular Biology Academia Sinica, Taiwan, R.O.C. Before joining the Faculty of NTHU, she taught in the Institute of Cellular and Molecular Biology, Taipei Medical College, from 1999 to 2000. Her current research interests include structural biology and bioinformatics.

Kun-Mao Chao was born in Tou-Liu, Taiwan, in 1963. He received the B.S. and M.S. degrees in computer engineering from National Chiao-Tung University, Taiwan, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from the Pennsylvania State University, University Park, USA, in 1993.

He is currently a professor of the Department of Computer Science and Information Engineering, National Taiwan University. From 1987 to 1989, he served in the ROC Air Force Headquarter as a system engineer. From 1993 to 1994, he worked as a postdoctoral fellow at the Center for Computational Biology, the Pennsylvania State University. In 1994, he was a visiting research scientist at the National Center for Biotechnology Information, National Institutes of Health, Bethesda, Maryland. Before joining the Faculty of NTUCSIE, he taught in the Department of Computer Science and Information Management, Providence University from 1994 to 1999, and the Department of Life Science, National Yang-Ming University from 1999 to 2002. His current research interests include algorithms and bioinformatics.

Dr. Chao is a member of Phi Tau Phi and Phi Kappa Phi.

Jia-Ming Chang received the B.S. and M.S. degrees in Department of Computer Science from National Tsing Hua University, Hsinchu, Taiwan, in 2000 and 2002, respectively. Recently, he is a software engineer in the Institute of Information Science of Academia Sinica. His research interests include design and analysis of algorithms and computational molecular biology.

Yu-Han Chiou received the B.S. degree from Soochow University, Taipei, Taiwan, in 1998, and M.S. degree in Computer Science and Information Engineering from National Hsing Hua University, Hsinchu, Taiwan, in 2002. He is now an engineer working on the development of the sound recognition technology in Panasonic Taiwan Laboratories Co., Ltd.

Chia-Mao Wu is now a Ph.D. student in Life Science Department, National Tsing Hua University, Taiwan. His current research interests include protein-protein interaction and cell biology.

Hao-Teng Chang is now a Ph.D. student in Life Science Department, National Tsing Hua University, Taiwan. His current research interests include protein-protein interaction, virus infection study, and heat shock stress.

Wei-I Chou is now a Ph.D. student in Life Science Department, National Tsing Hua University, Taiwan. His current research interests include protein-protein interaction and structural biology.