Structural Alignment of RNA with Complex Pseudoknot Structure

Thomas K.F. Wong[1], T.W. Lam[1], Wing-Kin Sung[2], and S.M. Yiu[1]

[1] Department of Computer Science, The University of Hong Kong, Hong Kong
{kfwong,twlam,smyiu}@cs.hku.hk
[2] School of Computing, National University of Singapore, Singapore
ksung@comp.nus.edu.sg

Abstract. The secondary structure of an ncRNA molecule is known to play an important role in its biological functions. Aligning a known ncRNA to a target candidate to determine the sequence and structural similarity helps in identifying de novo ncRNA molecules that are in the same family of the known ncRNA. However, existing algorithms cannot handle complex pseudoknot structures which are found in nature. In this paper, we propose algorithms to handle two types of complex pseudoknots: simple non-standard pseudoknots and recursive pseudoknots. Although our methods are not designed for general pseudoknots, it already cover all known ncRNAs in both Rfam and PseudoBase databases. A preliminary evaluation on our algorithms show that it is useful to identify ncRNA molecules in other species which are in the same family of a known ncRNA.

Keywords: Structural alignment, non-coding RNA, pseudoknots.

1 Introduction

A non-coding RNA (ncRNA) is a RNA molecule that does not translate into a protein. It has been shown to be involved in many biological processes [1,2,3]. The number of ncRNAs within the human genome was underestimated before, but recently some databases reveal over 212,000 ncRNAs [4] and more than 1,300 ncRNA families [5]. Large discoveries of ncRNAs and their families show the possibilities that ncRNAs may be as diverse as protein molecules [6]. Identifying ncRNAs is an important problem in biological study.

It is known that the secondary structure of an ncRNA molecule usually plays an important role in its biological functions. Some researches attempted to identify ncRNAs by considering the stability of secondary structures formed by the substrings of a given genome [16]. This method is not effective because a random sequence with high GC composition also allows an energetically favorable secondary structure [10]. A more promising direction is comparative approach which makes use of the idea that if a DNA region from which a RNA is transcribed has similar sequence and structure to a known ncRNA, then this region is likely to be an ncRNA gene whose corresponding ncRNA is in the same family of the known ncRNA. Thus, to locate ncRNAs in a genome, we can use a

S.L. Salzberg and T. Warnow (Eds.): WABI 2009, LNBI 5724, pp. 403–414, 2009.

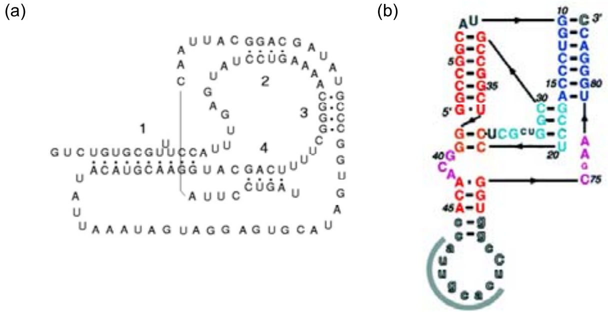

Fig. 1. (a): The secondary structure of RF00140 from Rfam 9.1 database [5]. Consider three base pairs: one from region 1, one from region 2 and one from region 4, they are mutually crossing each other (i.e. any two of them are crossing). (b): The secondary structure of self-cleaving ribozymes of hepatitis delta virus from [14] (i.e. RF00094 from Rfam 9.1 database).

known ncRNA as a query and search along the genome for substrings with similar sequence and structure to the query. The key of this approach is to compute the structural alignment between a query sequence with known structure and a target sequence with unknown structure. The alignment score represents their sequence and structural similarity. RSEARCH [11] and FASTR [12] belong to this category.

However, these tools do not support pseudoknots. Given two base pairs at positions (i, j) and (i', j'), where $i < j$ and $i' < j'$, pseudoknots are base pairs *crossing* each other, i.e. $i < i' < j < j'$ or $i' < i < j' < j$. In some studies, secondary structures including pseudoknots are found involved in some functions such as telomerase [7], catalytic functions [8], and self-splicing introns [9]. The presence of pseudoknots makes the problem computationally harder. Usually the large time complexity and considerable memory required for these algorithms make it impractical to search long pseudoknotted ncRNA along the genome.

Recently, Han et al. [15] developed PAL to solve the problem that supports secondary structures with standard pseudoknot of degree k and their algorithm runs in $O(kmn^k)$ where m is the length of the query sequence and n is the length of the target sequence. Their algorithm cannot handle more complex pseudoknot structures such as one with 3 base pairs mutually crossing each other (i.e. any two of them are crossing) as in Fig. 1(a) or the structure allowing recursive pseudoknots (i.e. pseudoknot/regular structures exist within another pseudoknot structure) as in Fig. 1(b). In Rfam 9.1 database [5], among 71 pseudoknotted families, 18 of them have complex pseudoknot structure. In the PseudoBase database [13], among 304 pseudoknot RNAs, 8 of them have complex pseudoknot structures. The small number may reflect the uncommon situation of having complex pseudoknots, but it may also reflect the difficulty of finding ncRNAs with complex pseudoknots due to the limitation of existing tools.

In this paper, we consider more complex pseudoknot structures which are found in nature. We define a class of pseudoknots called *simple non-standard pseudoknot* which allows some restricted cases with 3 base pairs mutually crossing each other. Our algorithm can apply to this complex structure in the same time complexity as Han's algorithm [15] for standard pseudoknot structure (i.e. $O(kmn^k)$ for degree k). Then, we propose an algorithm to handle recursive pseudoknot structure, our algorithm runs in $O(kmn^{k+1})$ if it is an *odd* structure or $O(kmn^{k+2})$ if it is an *even* structure. The definitions of odd and even structures for recursive pseudoknots will be given in Section 4.

Although our method is not designed for generic pseudoknots, we found that our method already cover all ncRNAs with complex pseudoknots in both Rfam 9.1 and PseudoBase databases. A preliminary experiment shows that our algorithms are useful in identifying ncRNAs from other species which are in the same family of a known ncRNA.

2 Pseudoknot Definitions

Let $A = a_1a_2\dots a_m$ be a length-m ncRNA sequence and M be the secondary structure of A. M is represented as a set of base pair positions. i.e. $M = \{(i,j)|1 \leq i < j \leq m, (a_i, a_j) \text{ is a base pair}\}$. Let $M_{x,y} \subseteq M$ be the set of base pairs within the subsequence $a_xa_{x+1}...a_y$, $1 \leq x < y \leq m$, i.e., $M_{x,y} = \{(i,j) \in M|x \leq i < j \leq y\}$. Note that $M = M_{1,m}$. We assume that there is no two base pairs sharing the same position, i.e., for any $(i_1,j_1),(i_2,j_2) \in M$, $i_1 \neq j_2$, $i_2 \neq j_1$, and $i_1 = i_2$ if and only if $j_1 = j_2$.

Definition 1. *$M_{x,y}$ is a* regular structure *if there does not exist two base pairs $(i,j),(k,l) \in M_{x,y}$ such that $i < k < j < l$ or $k < i < l < j$. Note that an empty set is also considered as a regular structure.*

A regular structure is one without pseudoknots. On the other hand, a standard pseudoknot of degree k allows certain types of pseudoknots. A structure is a standard pseudoknot of degree k if the RNA sequence can be divided into k consecutive regions (see Fig. 2(a)) such that base pairs must have end points in adjacent regions and base pairs that are in the same adjacent regions cannot cross each other. The formal definition is as follows.

Definition 2. *$M_{x,y}$ is a* standard pseudoknot of degree $k \geq 3$ *if* $\exists$ *a set of* pivot points $x_1, x_2, ..., x_{k-1}$ $(x = x_0 < x_1 < x_2 < ... < x_{k-1} < x_k = y)$ *that satisfy the following. Let $M_w(1 \leq w \leq k-1) = \{(i,j) \in M_{x,y}|x_{w-1} \leq i < x_w \leq j < x_{w+1}\}$. Note that we allow $j = x_k$ for M_{k-1} to resolve the boundary case.*
- *For each $(i,j) \in M_{x,y}$, $(i,j) \in M_w$ for some $1 \leq w \leq k-1$.*
- *$M_w(1 \leq w \leq k-1)$ is a regular structure.*

Note that a standard pseudoknot of degree 3 usually is referred as a *simple pseudoknot*. Now, we define a simple *non-standard* pseudoknot to include some structures with three base pairs crossing each other. For a simple non-standard

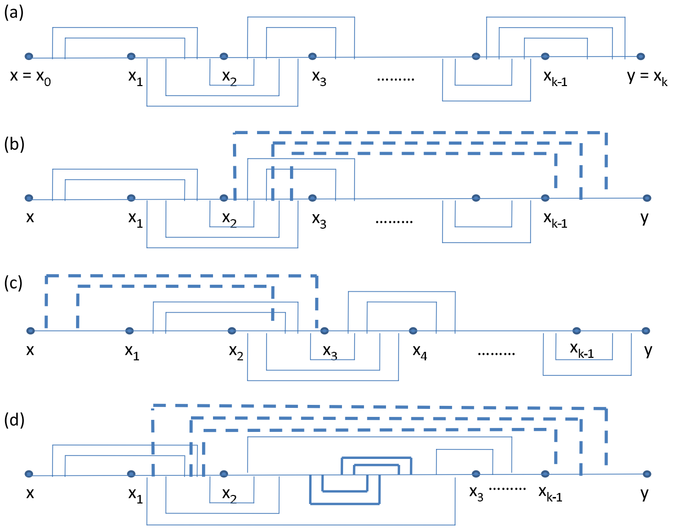

Fig. 2. (a)Standard pseudoknot of degree k. (b) Simple non-standard recursive pseudoknot of degree k (Type I). (c) Simple non-standard recursive pseudoknot of degree k (Type II). (d) Recursive pseudoknot (note the pseudoknot inside $[x_2, x_3]$).

pseudoknot of degree k, similar to a standard pseudoknot, the RNA sequence can be divided into k regions with the region at one of the ends (say, the right end) designated as the special region. Base pairs with both end points in the first $k-1$ regions have the same requirements as in a standard pseudoknot. And there is an extra group of base pairs that can start in one of the first $k-2$ regions and end at the last special region and again these pairs do not cross each other (see Fig. 2(b)). See the formal definition below.

Definition 3. *$M_{x,y}$ is a* simple non-standard pseudoknot of degree $k \geq 4$ (Type I) *if $\exists x_1, ..., x_{k-1}$ and t where $x = x_0 < x_1 < ... < x_{k-1} < x_k = y$ and* $1 \leq t \leq k-2$ *that satisfy the following. Let $M_w (1 \leq w \leq k-2) = \{(i,j) \in M_{x,y} | x_{w-1} \leq i < x_w \leq j < x_{w+1}\}$. Let $X = \{(i,j) \in M_{x,y} | x_{t-1} \leq i < x_t, x_{k-1} \leq j \leq y\}$.*

- *For each $(i,j) \in M_{x,y}$, either $(i,j) \in M_w (1 \leq w \leq k-2)$ or $(i,j) \in X$.*
- *M_w and X are regular structures.*

Type II simple non-standard pseudoknots (see Fig. 2(c)) are symmetric to Type I simple non-standard pseudoknots with the special region on the left end. In the rest of the paper, we only consider Type I simple non-standard pseudoknots and simply refer it as simple non-standard pseudoknots.

Lastly, we define what a recursive pseudoknot is (see Fig. 2(d)).

Definition 4. *$M_{x,y}$ is a recursive pseudoknot of degree $k \geq 3$ if $M_{x,y}$ is either regular, standard pseudoknot of degree k or simple non-standard pseudoknot of* *degree k (if $k \geq 4$), or $\exists a_1, b_1, ..., a_s, b_s (x \leq a_1 < b_1 < ... < a_s < b_s \leq y)$ that satisfy the followings. Each M_{a_i,b_i} is called a* recursive region.

- *M_{a_i,b_i}, for $1 \leq i \leq s$, is a recursive pseudoknot of degree $\leq k$.*
- *$(M_{x,y} - \bigcup_{1 \leq i \leq s} M_{a_i,b_i})$ is either regular structure, standard pseudoknot of degree $\leq k$ or simple non-standard-pseudoknot of degree $\leq k$.*

3 Algorithm for Simple Non-standard Pseudoknots

3.1 Structural Alignment

Let $S[1...m]$ be a query sequence with known secondary structure M, and $T[1...n]$ be a target sequence with unknown secondary structure. S and T are both sequences of {A,C,G,U}. A structural alignment between S and T is a pair of sequences $S'[1...r]$ and $T'[1...r]$ where $r \geq m, n$, S' is obtained from S and T' is obtained from T with spaces inserted to make both of the same length. A space cannot appear in the same position of S' and T'. The score of the alignment, which determines the sequence and structural similarity between S' and T', is defined as follows [12].

$$score = \sum_{i=1}^{r} \gamma(S'[i], T'[i]) + \sum_{\substack{i,j \text{ s.t. } \eta(i),\eta(j) \in M, \\ S'[i],S'[j],T'[i],T'[j] \neq \text{`\_'}}} \delta(S'[i], S'[j], T'[i], T'[j]) \qquad (1)$$

where $\eta(i)$ is the corresponding position in S according to the position i in S'; $\gamma(t_1, t_2)$ and $\delta(x_1, y_1, x_2, y_2)$ where $t_1, t_2 \in \{A, C, G, U, \text{`\_'}\}$ and $x_1, x_2, y_1, y_2 \in \{A, C, G, U\}$, are scores for character similarity and for base pair similarity, respectively. The problem is to find an alignment to maximize the score.

3.2 Substructure of Simple Non-standard pseudoknot

We solve the problem using dynamic programming. The key is to define a substructure to enable us to find the solution recursively. For ease understanding of what a substructure is, we draw the pseudoknot structure using another approach (see Fig. 3).

We use simple non-standard pseudoknots with degree 4 for illustration. The result can be easily extended to general k. Fig. 3(b) shows the same pseudoknot structure as in Fig. 3(a). By drawing the pseudoknot structure this way, the base pairs can be drawn without crossing and can be ordered from the top to bottom. According to this ordering, we can define a substructure based on four points on the sequence (see Fig. 3(c) in which the substructure is highlighted in bold) such that all base pairs are either with both end points inside or outside the substructure. Note that in Fig. 3(c), $t = 1$ (t is odd), if $t = 2$ (t is even), we have to use a slightly different definition for substructures, otherwise base pairs cannot be ordered from top to bottom without crossing each other (see Fig. 3(d) and (e). Note that the two base pairs that cross in Fig. 3(d) is due to the way

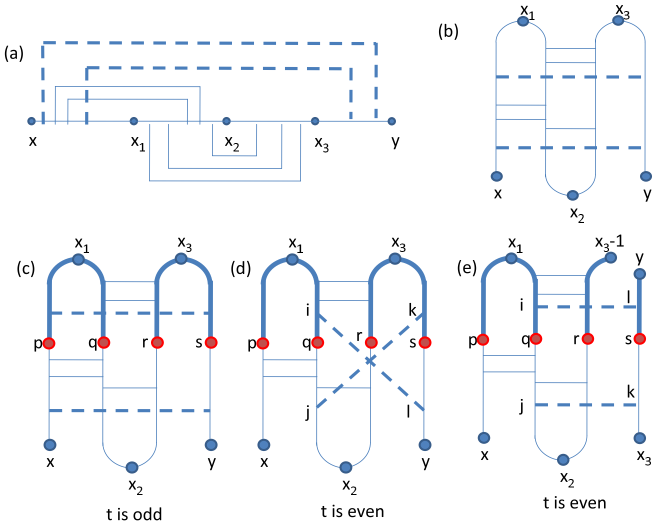

Fig. 3. Substructure of a simple non-standard pseudoknot

we draw the pseudoknot, they do not actually cross each other. i.e. basepairs (i,l) and (j,k) do not form a pseudoknot.). These are the only cases we need to consider.

Now, we formally define what a substructure is. Let $S[x..y]$ be an RNA sequence with known simple non-standard pseudoknot structure M of degree 4. Note that x_1, x_2, x_3 and t are known. Let $v = (p, q, r, s)$ be a quadruple with $x \leq p < x_1 \leq q < x_2 \leq r \leq x_3 < s \leq y$. If t is odd, define the *subregion* $R_{odd}(S, v) = [p, q] \cup [r, s]$. Otherwise, define the subregion $R_{even}(S, x_3, v) = [p, q] \cup [r, x_3 - 1] \cup [s, y]$. Note that x_3 is not a parameter, but a fixed value for S. Let $Struct(R_x) = \{(i, j) \in M | i, j \in R_x\}$ where R_x is a subregion.

We say that a subregion R_x defines a valid substructure ($Struct(R_x)$) of M if there does not exist $(i, j) \in M$ such that one endpoint of (i, j) is in R_x and the other is outside the region. Obviously, $Struct(R_x)$ is also a simple non-standard pseudoknot structure.

3.3 Dynamic Programming

Let $S[1, m]$ be the query sequence with known structure M and $T[1, n]$ be the target sequence with unknown structure. Note that the pivot points x_1, x_2, x_3 and t for S is known. We can apply the definitions of R_{odd} and R_{even} to T. If t is odd, for any $v' = (e, f, g, h)$ such that $1 \leq e < f < g < h \leq n$, we define the subregion $R_{odd}(T, v') = [e, f] \cup [g, h]$. If t is even, for any $v' = (e, f, g, h)$ and x'_3 such that $1 \leq e < f < g < x'_3 \leq h \leq n$, we define the subregion $R_{even}(T, x'_3, v') = [e, f] \cup [g, x'_3 - 1] \cup [h, n]$. Note that since the structure of T is unknown, x'_3 is a parameter.

Define $C(R_x, R_y)$ be the score of the optimal alignment between a subregion R_x in S with substructure $Struct(R_x)$ and a subregion R_y in T. The score of the optimal alignment between S and T can be obtained as follows. If t is odd, setting $v^* = (1, x_2 - 1, x_2, m)$ includes the whole query sequence S, the entry $\max_{x'_2}\{C(R_{odd}(S, v^*), R_{odd}(T, v' = (1, x'_2 - 1, x'_2, n)))\}$ provides the answer. On the other hand, if t is even, setting $v^* = (1, x_2 - 1, x_2, x_3)$, the entry $\max_{x'_2} \max_{x'_3 > x'_2}\{C(R_{even}(S, x_3, v^*), R_{even}(T, x'_3, v' = (1, x'_2 - 1, x'_2, x'_3)))\}$ provides the optimal score.

The value of $C(R_x, R_y)$ can be computed recursively. Assume that t is odd. Let $R_x = R_{odd}(S, (p, q, r, s))$ and $R_y = R_{odd}(T, (e, f, g, h))$. If (p, q) is a base pair in $Struct(R_x)$, there are four cases to consider. Case 1: MATCH$_{both}$ - aligning the base pair (p, q) of S with (e, f) of T; Case 2: MATCH$_{single}$ - aligning only one of the bases in (p, q) with the corresponding base in (e, f); Case 3: INSERT - insert a space on S; Case 4: DELETE - delete the base-pair (p, q) from S. Lemma 1 summarizes these cases.

The other cases, (q, r) is a base pair or (p, s) is a base pair, are similar. Note that if more than one such base pair exists (e.g. both (q, r) and (p, s) are base pairs), we only need to follow the recursion on one of the pairs. However, you cannot pick any of them in an arbitrary manner, otherwise, when we fill the dynamic programming table, we need to fill all entries for all possible subregions of S. We will address this issue in the later part of this section.

Lemma 1. *Let $v = (p, q, r, s)$ and $v' = (e, f, g, h)$. Let t be odd. And $R_x = R_{odd}(S, v)$, $R_y = R_{odd}(T, v')$. If (p, q) is a base pair, then* $C(R_x, R_y) = \max$

$$\begin{cases}
//MATCH_{both} \\
C(R_{odd}(S, (p+1, q-1, r, s)), R_{odd}(T, (e+1, f-1, g, h))) \\
\quad + \gamma(S[p], T[e]) + \gamma(S[q], T[f]) + \delta(S[p], S[q], T[e], T[f]); \\
//MATCH_{single} \\
C(R_{odd}(S, (p+1, q-1, r, s)), R_{odd}(T, (e+1, f, g, h))) + \gamma(S[p], T[e]) + \gamma(S[q], \text{`\_'}), \\
C(R_{odd}(S, (p+1, q-1, r, s)), R_{odd}(T, (e, f-1, g, h))) + \gamma(S[p], \text{`\_'}) + \gamma(S[q], T[f]); \\
//INSERT \\
C(R_{odd}(S, (p, q, r, s)), R_{odd}(T, (e+1, f, g, h))) + \gamma(\text{`\_'}, T[e]), \\
C(R_{odd}(S, (p, q, r, s)), R_{odd}(T, (e, f-1, g, h))) + \gamma(\text{`\_'}, T[f]), \\
C(R_{odd}(S, (p, q, r, s)), R_{odd}(T, (e, f, g+1, h))) + \gamma(\text{`\_'}, T[g]), \\
C(R_{odd}(S, (p, q, r, s)), R_{odd}(T, (e, f, g, h-1))) + \gamma(\text{`\_'}, T[h]), \\
//DELETE \\
C(R_{odd}(S, (p+1, q-1, r, s)), R_{odd}(T, (e, f, g, h))) + \gamma(S[p], \text{`\_'}) + \gamma(S[q], \text{`\_'})
\end{cases}$$

On the other hand, if none of these are base pairs, assume that $p + 1 < x_1$ and $S[p]$ is a single base, then we can compute $C(R_x, R_y)$ recursively according to another three cases. Case 1: Match - aligning $S[p]$ with $T[e]$; Case 2: INSERT - insert a space on S; Case 3: Delete - delete $S[p]$.

Lemma 2. *Let $v = (p, q, r, s)$ and $v' = (e, f, g, h)$. Let t be odd. And $R_x = R_{odd}(S, v)$, $R_y = R_{odd}(T, v')$. If $p + 1 < x_1$ and $S[p]$ is a single base, then* $C(R_x, R_y) = \max$

$$\begin{cases} C(R_{odd}(S,(p+1,q,r,s)), R_{odd}(T,(e+1,f,g,h))) + \gamma(S[p],T[e]) \ //\mathit{MATCH} \\ //\mathit{INSERT\text{: } same\ as\ the\ one\ defined\ in\ Lemma\ 1} \\ C(R_{odd}(S,(p+1,q,r,s)), R_{odd}(T,(e,f,g,h))) + \gamma(S[p], \text{'\_'}) \ //\mathit{DELETE} \end{cases}$$

For t is even, we consider whether (p, q),(q, r), and (q, s) are base pairs in $Struct(R_x)$ and we need to consider all possible cases for x'_3 since the structure of T is unknown (i.e.,the pivot points are unknown).

To fill the dynamic programming table, not all entries for all possible subranges of S needs to be filled. For any given subregion $v = (p, q, r, s)$ in S, we first define $pair_{min}(v)$ and $single_{min}(v)$ as follow. If there exists a set of base pairs, say $\{(i_1, j_1), ..., (i_d, j_d)\}$, such that all $i_k, j_k (1 \le k \le d)$ equals to p (if $x \le p < x_1$), q (if $x_1 \le q < x_2$), r (if $x_2 \le r < x_3$) or s (if $x_3 \le s \le y$), then $pair_{min}(v)$ is the pair with minimum value of i. Also, if there exists a set of single bases (i.e. the positions which do not belong to any base pair), say $\{u_1, ..., u_d\}$, such that all $u_k (1 \le k \le d)$ equals to p (if $x \le p < x_1$), q (if $x_1 \le q < x_2$), r (if $x_2 \le r < x_3$) or s (if $x_3 \le s \le y$), then $single_{min}(v)$ is the one with minimum value.

Now, we define a function $\zeta(v)$ to determine subregions in S, for which we need to fill the corresponding C entires.

Case 1. If $(i, j) = pair_{min}(v)$ exists, then

$$\zeta(v) = \begin{cases} (p+1, q-1, r, s), & \text{if } (i,j) = (p,q) \\ (p, q-1, r+1, s), & \text{if } (i,j) = (q,r) \\ (p+1, q, r, s-1), & \text{if } (i,j) = (p,s) \text{ i.e. } t \text{ is odd} \\ (p, q-1, r, s+1), & \text{if } (i,j) = (q,s) \text{ i.e. } t \text{ is even} \end{cases} \tag{2}$$

Case 2. If $pair_{min}(v)$ does not exist, then $u = single_{min}(v)$ should exist and

$$\zeta(v) = \begin{cases} (p+1, q, r, s), & \text{if } u = p \\ (p, q-1, r, s), & \text{if } u = q \\ (p, q, r+1, s), & \text{if } u = r \\ (p, q, r, s-1), & \text{if } u = s \text{ and } t \text{ is odd} \\ (p, q, r, s+1), & \text{if } u = s \text{ and } t \text{ is even} \end{cases} \tag{3}$$

It is obvious that if v defines a subregion with a valid substructure, $\zeta(v)$ also defines a valid substructure. For t is odd, let $v* = (1, x_2 - 1, x_2, m)$. We only need to fill in the entries for C provided v can be obtained from $v*$ by applying ζ function repeatedly. If t is even, let $v* = (1, x_2 - 1, x_2, x_3)$. Intuitively, ζ guides which recursion formula to use. And there are only $O(m)$ such v values. The following lemma summarizes the time complexity for this algorithm.

Lemma 3. *For any sequence $S[1..m]$ with simple non-standard pseudoknot of degree 4 and any sequence $T[1..n]$, let c be the max length of $[x'_3, n]$, the optimal alignment score between $S[1..m]$ and $T[1..n]$ can be computed in $O(cmn^4)$.*

Note that the factor c is only needed when t is even due to the extra parameter x'_3. We examined all sequences in Rfam and PseudoBase, we found that usually the length of the final segment c is short (< 15) and the average length is only 5.4 with most of the cases having lengths from 5 to 7. So, we can assume that c is a constant. The algorithm can be extended to simple non-standard pseudoknot of degree k easily.

Theorem 1. *For any sequence $S[1..m]$ with simple non-standard pseudoknot of degree k and any sequence $T[1...n]$, the optimal alignment score between $S[1..m]$ and $T[1..n]$ can be computed in $O(kmn^k)$.*

4 Algorithm for Recursive Pseudoknot

We use the recursive pseudoknot of degree 4 to illustrate the algorithm. The approach can be easily extended to general k. Let $S[1..m]$ be the query sequence with recursive pseudoknot structure M. Recall the definition of a recursive pseudoknot. There can be disjoint recursive regions, namely $M_{a_1,b_1}, \ldots, M_{a_s,b_s}$, in M. By removing all these recursive regions, the remaining structure $M - (M_{a_1,b_1} \cup \cdots \cup M_{a_s,b_s})$ together with the remaining sequence $S[1..a_1-1]S[b_1+1..a_2-1] \ldots S[b_s+1..m]$ are referred as level-0. For each removed recursive region M_{a_i,b_i}, we can apply the same procedure to define level-1, level-2, ..., level-ℓ structures (see Fig. 4 for an example). In this section, we assume all the recursive regions M_{a_i,b_i} in all levels including $M_{1,m}$ after removing the next-level substructure are simple non-standard pseudoknots or regular structure. Among all these recursive regions with degree-4, if there exists one of them for which the value of t is even, then the whole recursive pseudoknot structure is referred as an *even* structure, otherwise, it is an *odd* structure.

Let $T[1..n]$ be the target sequence. Define $H[a_i, b_i, x', y']$ be the score of the optimal alignment between the recursive region $S[a_i, b_i]$ with structure M_{a_i,b_i}

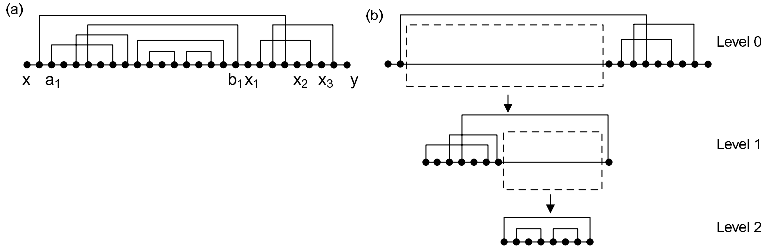

Fig. 4. An example showing that recursive pseudoknot of degree 4 can be divided into levels: Level 0 is a simple non-standard pseudoknot of degree 4; Level 1 is another simple non-standard pseudoknot; Level 2 is a regular structure

and $T[x'..y']$, where $1 \le x' < y' \le n$. We now show how to compute the score of the optimal alignment between S and T recursively. We assume that level-0 structure is a simple non-standard pseudoknot and consider t is odd. Let $v = (p, q, r, s)$ be a quadruple that defines a substructure of S. Let $S[p..y_p]$ be a recursive region. The following lemma shows how to compute $C(R_x, R_y)$ where $R_x = R_{odd}(S, v)$ and $R_y = R_{odd}(T, v')$.

Lemma 4. *Let $v = (p, q, r, s)$ and $v' = (e, f, g, h)$. Assume that t is odd. $R_x = R_{odd}(S, v)$ and $R_y = R_{odd}(T, v')$. If $S[p..y_p]$ is a recursive region, then*

$$C(R_x, R_y) = \max \begin{cases} //MATCH \\ \max_{e \le w \le f}\{C(R_{odd}(S, (y_p + 1, q, r, s)), R_{odd}(T, (w + 1, f, g, h))) + H(p, y_p, e, w)\} \\ //INSERT \\ \text{ same as INSERT defined in Lemma 1} \\ //DELETE \\ C(R_{odd}(S, (y_p + 1, q, r, s)), R_{odd}(T, (e, f, g, h))) + \sum_{p \le w \le y_p} \gamma(S[w], \text{'\_'}) \end{cases}$$

Other cases, such as $S[x_q..q]$ or $S[r..y_r]$ or $S[x_s..s]$ is a recursive region, are handled in a similar way. Again, we need to determine for which subregions in S, we need to fill in the corresponding C entries. So, we enhance ζ function as follows.

Consider a quadruple $v = (p, q, r, s)$ in a region $S[x...y]$ where the structure is a simple non-standard pseudoknot of degree 4 if all the next-level subregions inside are excluded. Let us define subregion$_{min}(v)$ as follows: if there exists a set of next-level subregions, say $\{[i_1, j_1], ..., [i_d, j_d]\}$ where $x \le i_k < j_k \le y$ for all $1 \le k \le d$ such that either i_k or j_k equals to p (if $x \le p < x_1$), q (if $x_1 \le q < x_2$), r (if $x_2 \le r < x_3$) or s (if $x_3 \le s \le y$), then let subregion$_{min}(v)$ be the region with minimum value of i. We add the following case to ζ function. Note that the t value refers to the structure for $S[x..y]$ excluding all next-level subregions.

Case 0 of $\zeta(v)$: If $[i, j] = \text{subregion}_{min}(v)$ exists, then

$$\zeta(v) = \begin{cases} (j + 1, q, r, s), & \text{if } i = p \\ (p, i - 1, r, s), & \text{if } j = q \\ (p, q, j + 1, s), & \text{if } i = r \\ (p, q, r, i - 1), & \text{if } j = s \text{ //i.e. } t \text{ is odd} \\ (p, q, r, j + 1), & \text{if } i = s \text{ //i.e. } t \text{ is even} \end{cases} \tag{4}$$

It remains to show how to compute $H()$. We start from the lowest level structure. Let $S[a..b]$ be such a subregion with structure $M[a..b]$. Assume that $M[a..b]$ is a simple non-standard pseudoknot. In a brute-force manner, for any $1 \le x' < y' \le n$, we can apply the algorithm in Section 3 to compute $H[a, b, x', y']$. However, this may takes $O(\alpha n^6)$ time, where $\alpha = (b - a)$. In fact, we are able to speed up the computation so that $H()$ can be computed in $O(\alpha n^4)$ time if t is odd and $O(\alpha n^5)$ if t is even. Note that once a subregion is considered to compute $H()$ function, in subsequent steps, we do not need to process that subregion again.

The following theorem shows the main result of this section (the full details will be given in the full paper) assuming that inside the recursive pseudoknot, there are some simple non-standard pseudoknots.

Theorem 2. *To compute the optimal alignment score between a query sequence* $S[1...m]$ *with recursive pseudoknot of degree* k (≥ 4) *and a target sequence* $T[1...n]$*, it can be done with the following time complexity.*

$$\textit{time complexity} = \begin{cases} O(kmn^{k+1}), & \textit{if it is an odd structure;} \\ O(kmn^{k+2}), & \textit{if it is an even structure;} \end{cases}$$

Our algorithm for recursive pseudoknot can be easily adapted to cases in which the recursive pseudoknot can have a mix of regular structure, standard pseudoknot of degree k, and simple non-standard pseudoknot of degree k.

5 Preliminary Evaluation

We selected two families in Rfam 9.1 database to test the effectiveness of our algorithms. RF00094 is a family with members having recursive pseudoknots and RF00140 is another with members having simple non-standard pseudoknots. Both structures of RF00094 and RF00140 are of degree 4 (see Fig. 1(a) and (b)). In the experiment, we select a known member of the family, scan the genome of another species and check whether the known members of that species can be identified. Following the evaluation method of [15], since there is no existing software which can perform structural alignment for complex pseudoknot structure, we compared the performance of our programs with BLAST. We use default parameters for BLAST except the wordsize is set to 7. Fig. 5 shows the result for family RF00094. The two regions of known members have the highest scores from our program. Although one of their BLAST score is the highest, another one is not quite distinguishable from others. For RF00140, there are two known members in this genome region and our program identifies 4 possible candidates

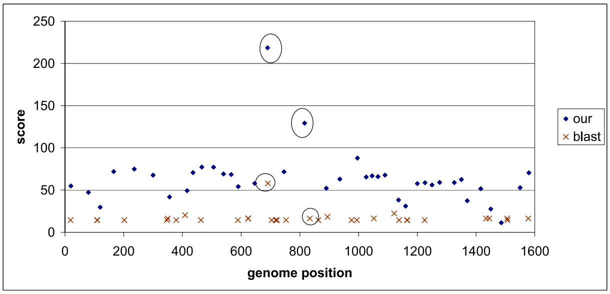

Fig. 5. Comparison of resulting scores from our program and BLAST when scanning the whole genome (AB118824) for family RF00094 using AB037948/685-775 as the query sequence. The circled points are the known members of the same family in genome AB118824.

including these two known members. On the other hand, BLAST resulted in three regions but none of them matches with any of known members.

6 Conclusions

In the paper, we provided the first algorithms to handle structural alignment of RNA with two complex pseudoknot structures, recursive pseudoknots and simple non-standard pseudoknots. Further directions include speeding up these algorithms and considering other more complicated pseudoknot structures.

References

1. Frank, D.N., Pace, N.R.: Ribonuclease P: unity and diversity in a tRNA processing ribozyme. Annu. Rev. Biochem. 67, 153–180 (1998)
2. Nguyen, V.T., et al.: 7SK small nuclear RNA blinds to and inhibits the activity of CDK9/cyclin T complexes. Nature 414, 322–325 (2001)
3. Yang, Z., et al.: The 7SK small nuclear RNA inhibits the CDK9/cyclin T1 kinase to control transcription. Nature 414, 317–322 (2001)
4. Liu, C., et al.: NONCODE: an integrated knowledge database of non-coding RNAs. NAR 33(Database issue), D112–D115 (2005)
5. Griffiths-Jones, S., et al.: Rfam: an RNA family database. NAR 31(1), 439–441 (2003), `http://www.sanger.ac.uk/Software/Rfam/`
6. Eddy, S.: Non-coding RNA genes and the modern RNA world. Nature Reviews in Genetics 2, 919–929 (2001)
7. Hen, J., Greider, C.W.: Functional analysis of the pseudoknot structure in human telomerase RNA. PNAS 102(23), 8080–8085 (2005)
8. Dam, E., Pleij, K., Draper, D.: Structural and functional aspects of RNA pseudoknots. Biochemistry 31(47), 11665–11676 (1992)
9. Adams, P.L., Stahley, M.R., Kosek, A.B., Wang, J., Strobel, S.A.: Crystal structure of a self-splicing group I intron with both exons. Nature 430, 45–50 (2004)
10. Rivas, E., Eddy, S.: Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. Bioinformatics 16(7), 583–605 (2000)
11. Klei, R.J., Eddy, S.R.: RSEARCH: Finding homologs of single structured RNA sequences. BMC Bioinformatics 4(1), 44 (2003)
12. Zhang, S., Hass, B., Eskin, E., Bafna, V.: Searching genomes for noncoding RNA using FastR. IEEE/ACM TCBB 2(4) (2005)
13. van Batenburg, F.H.D., Gultyaev, A.P., Pleij, C.W.A., Ng, J., Oliehoek, J.: Pseudobase: a database with RNA pseudoknots. NAR 28(1), 201–204 (2000)
14. Ferre-D'Amare, A.R., Zhou, K., Doudna, J.A.: Crystal structure of a hepatitis delta virus ribozyme. Nature 395, 567–574 (1998)
15. Han, B., Dost, B., Bafna, V., Zhang, S.: Structural Alignment of Pseudoknotted RNA. JCB 15(5), 489–504 (2008)
16. Le, S.Y., Chen, J.H., Maizel, J.: Efficient searches for unusual folding regions in RNA sequences. In: Structure and Methods: Human Genome Initiative and DNA Recombination, vol. 1, pp. 127–130 (1990)