JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 19, Number 4, 2012
© Mary Ann Liebert, Inc.
Pp. 365–378
DOI: 10.1089/cmb.2010.0052

Structural Alignment of RNA with Triple Helix Structure

THOMAS K.F. WONG and S.M. YIU

ABSTRACT

Structural alignment is useful in identifying members of ncRNAs. Existing tools are all based on the secondary structures of the molecules. There is evidence showing that tertiary interactions (the interaction between a single-stranded nucleotide and a base-pair) in triple helix structures are critical in some functions of ncRNAs. In this article, we address the problem of structural alignment of RNAs with the triple helix. We provide a formal definition to capture a simplified model of a triple helix structure, then develop an algorithm of $O(mn^3)$ time to align a query sequence (of length m) with known triple helix structure with a target sequence (of length n) with an unknown structure. The resulting algorithm is shown to be useful in identifying ncRNA members in a simulated genome.

Key words: algorithms, non-coding RNA, structural alignment, triple helix.

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 (Frank and Pace, 1998, Nguyen et al., 2001, Yang et al., 2001). Identifying ncRNAs is an important problem in biological study. It is known that the structure of an ncRNA molecule usually plays an important role in its biological functions. Some research attempted to identify ncRNAs by considering the stability of secondary structures formed by the substrings of a given genome (Le et al., 1990). This method is not effective because a random sequence with high GC composition also allows an energetically favorable secondary structure (Rivas and Eddy, 2000). A more promising direction is a comparative approach, which makes use of the idea that if a DNA region from which a RNA is transcribed has sequence and structure similar 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 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.

Recently, a number of methods have been developed to compute the structural alignment between a query sequence with known structure and a target sequence with unknown structure. RSEARCH (Klein and Eddy, 2003) and FASTR (Zhang et al., 2005) are two software programs designed for the query sequence with regular structure. Matsui et al. (2005), Han et al. (2008), and Wong et al. (2009) also developed algorithms to solve the structural alignment problem that supports different types of pseudoknot structures.

Department of Computer Science, The University of Hong Kong, Hong Kong.

All these methods have an assumption that each nucleotide can interact with at most one nucleotide in the ncRNA. However, if tertiary interaction is considered, this assumption may not hold.

Triple helix structure considers tertiary interaction in the ncRNA. Inside the triple helix, some single-stranded nucleotides will form hydrogen bonds with nucleotides in base pairs. Figure 1 shows an example of a triple helix structure. Triple helix structure exists in yeast and human telomerase and the one in human telomerase also conserves in all vertebrates (Qiao and Cech, 2008, Chen and Greider, 2005, Theimer et al., 2005). Telomerase is responsible for adding specific sequence repeats to the ends of chromosomes and is important for maintaining telomere length and chromosome stability in stem cells, germline cells, and cancer cells (Chen and Greider, 2005). Qiao and Cech (2008) showed that breaking the tertiary interaction inside the triple helix structure of the telomerase will deteriorate the corresponding activity *in vitro* and shorten the telomere *in vivo*. On the other hand, triple helix structure also appears in the pseudoknot active in ribosomal frameshifting (Su et al., 1999). Frameshifting makes a shift in reading frames, causes the transcription process to skip the stopping codon and produces a single fusion protein. The tertiary interation between single-stranded nucleotides in the loop and base pairs in the stem (i.e., loop-stem interaction) is found to be essential for efficient frameshifting (Chen et al., 1995).

Since the tertiary interaction between single-stranded nucleotide and base pair (in short, we use ‘‘tertiary interaction’’ in the rest of the article) in the triple helix structure is important to the function of ncRNA, it is better to consider these tertiary interactions when performing structural alignment. In this article, we consider the structural alignment problem for triple helix structure. Based on the known examples of triple helix, we observe that one base-pair may interact with more than one single-stranded nucleotide, and one single-stranded nucleotide may also interact with more than one base-pair. Also, these tertiary interactions usually occur over a simple pseudoknot structure. Along with other observations, we try to provide a formal definition to capture the structure of a triple helix. We refer this as a *standard triple helix structure*. Then, we develop a structural alignment algorithm to align a query sequence with known triple helix structure and a target sequence with an unknown structure. Our alignment algorithm runs in $O(mn^3)$ time, which is the same as the time complexity of the alignment algorithm for simple pseudoknot structure described by Han et al. (2008), although we also consider the tertiary interactions inside the pseudoknot.

We implemented the algorithm and evaluated it based on a simulated genome. The results show that it is effective in identifying ncRNAs from the genome which are in the same family of a known ncRNA with triple helix structure. We remark that this is the first attempt to consider tertiary interactions in the structural alignment. The model we propose will not be the ultimate model for all triple helix structures. A more accurate model should be developed after more triple helix structures are known and studied.

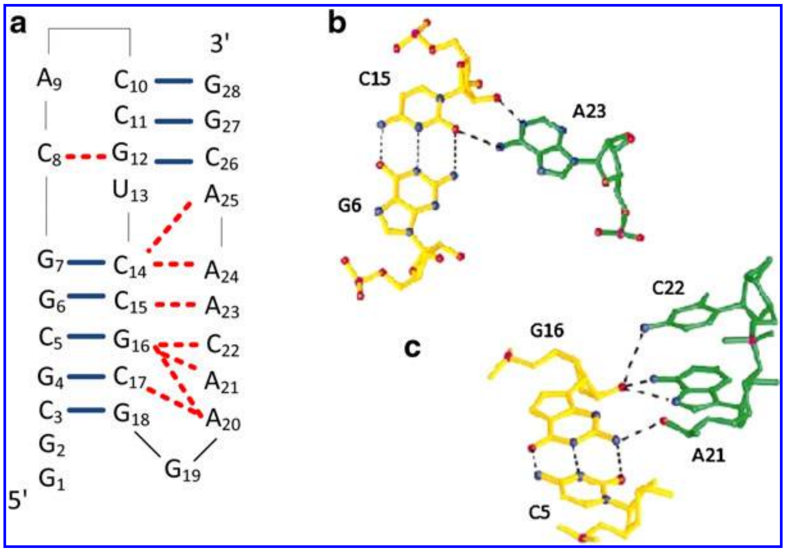


FIG. 1. **(a)** Triple helix in beet western yellow virus pseudoknot (Su et al., 1999). Blue lines represent the secondary structure; red lines represent the tertiary interactions between single-stranded nucleotides (according to the secondary structure) and base pairs. **(b, c)** Detailed view of some tertiary interactions in the structure (Su et al., 1999): A single-stranded nucleotide (A23) interacts with a base pair (G6,C15) (b); and a base pair (C5,G16) interacts with two single-stranded nucleotides A21 and C22 (c).

2. DEFINITIONS

Standard triple helix: Figure 1b, c shows the interactions between nucleotides inside a triple helix structure of a beet western yellow virus pseudoknot which is active in ribosomal frameshifting. We analyzed the available triple helix structures (Chastain and Tinoco, 1992; Chen and Greider, 2005; Qiao and Cech, 2008; Su et al., 1999) and came up with the following observations or assumptions for a simplified abstract model for such a structure:

1. When a single-stranded nucleotide interacts with a base-pair, the single-stranded nucleotide may interact with one of the nucleotides in the base pair such as the example in Figure 1b or interact with both nucleotides in the base pair. For simplicity, we regard both cases equivalent and refer it as the interaction between the single-stranded nucleotide with the base pair. We denote it as $(i, j) * k$ where (i, j) is the base pair and k denotes the single-stranded nucleotide.
2. A base pair may interact with more than one single-stranded nucleotide such as in Figure 1c, where (C_5, G_{16}) interacts with both A_{21} and C_{22}. On the other hand, a single-stranded nucleotide may also interact with more than one base pair.
3. The underlying secondary structure of the base pairs is usually a simple pseudoknot or simple pseudoknot like. This is probably due to the stable nature of the simple pseudoknot structure.
4. Based on the underlying simple pseudoknot (Han et al., 2008) like structure, base pairs are divided into two groups (see the formal definition below). Each group spans a region in the sequence. The single-stranded nucleotides that interact with base pairs of a group are usually outside of its spanned region.
5. If the triple helix structure is drawn as in Figure 1a, that is, besides an edge between each base pair, a conceptual interaction edge is drawn from a single-stranded nucleotide to the closest nucleotide of the base pair that it interacts with, it is assumed that there is no crossing in all edges.

There are probably exceptions that do not follow our observations and assumptions, but not yet discovered. However, it may be reasonable to make these assumptions for the time being as a starting point for studying the structural alignment with triple helix structure. The standard triple helix structure is formally defined as follows.

Let $A = a_1 a_2 \ldots a_m$ be a length-m ncRNA sequence. Let M be underlying secondary structure of A i.e. $M = \{(i, \ j) | 1 \leq i < j \leq m, \ (a_i, \ a_j) \text{ is a base pair}\}$. Let P be the tertiary interactions of A i.e. $P = \{(i, j) * k | (i, j) \in M, a_k \text{ is a single-stranded nucleotide and interacts with } (a_i, a_j)\}$. Then, $H = (M, P)$ is referred as the triple helix structure of A.

The secondary structure still obeys the rule that 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 $j_1 = j_2$. However, the tertiary interactions do not follow this rule (based on Observations (2) and (3)), i.e., for any $(i_1, j_1) * k_1, (i_2, j_2) * k_2 \in P$, if $i_1 = i_2$ and $j_1 = j_2$, it does not imply $k_1 = k_2$, and also, if $k_1 = k_2$, it does not imply $i_1 = i_2$ and $j_1 = j_2$.

$H = (M, P)$ is a *standard triple helix structure*, as illustrated in Figure 2a, if $\exists x_1, x_2 (1 \leq x_1 < x_2 \leq m)$ that satisfy the following. Let $R_1 = \{(i, j) \in M | 1 \leq i < x_1 \leq j < x_2\}$ and $R_2 = \{(i, j) \in M | x_1 \leq i < x_2 \leq j \leq m\}$.

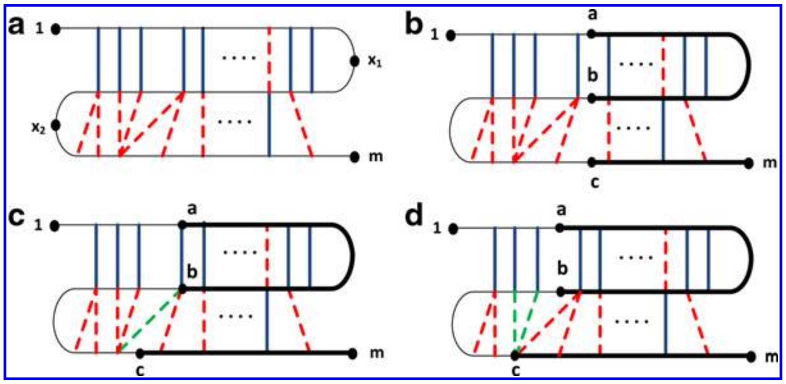


FIG. 2. **(a)** A standard triple helix structure. **(b–d)** A subregion $Region(a, b, c) = [a \ldots b] \cup [c \ldots m]$ is valid if all positions of base pairs and tertiary interactions are either inside or outside (b); or if a base pair of a tertiary interaction is inside the subregion but the single-stranded base is not, then the base pair is at the end point of the subregion (c); or if a single-stranded base of a tertiary interaction is inside the subregion but the base pair is not, then the single-stranded base is at the end point of the subregion (d).

- For any two base pairs $(i_1, j_1), (i_2, j_2) \in R_k$, $k = 1$ or 2, either $i_1 < i_2 < j_2 < j_1$ or $i_2 < i_1 < j_1 < j_2$. That is, the base pairs in the same group do not cross.
- $R_1 \cup R_2 = M$. (R_1, R_2 form the simple pseudoknot structure)
- For any $(i, j) * k \in P$, , if $(i, j) \in R_1$, then $x_2 \leq k \leq m$ and $\nexists$ $(i', j') \in R_2$ such that $j \leq i' \leq k \leq j'$ or $i' \leq j \leq j' \leq k$. This is to make sure that k is from an outside region of R_1 and there does not exist base pairs in R_2 that cross with the tertiary interaction (Observations (5) and (6)). Similarly, if $(i, j) \in R_2$, then $1 \leq k < x_1$ and $\nexists$ $(i', j') \in R_1$ such that $k \leq i' \leq i \leq j'$ or $i' \leq k \leq j' \leq i$.
- For any $(i_1, j_1) * k_1, (i_2, j_2) * k_2 \in P$, if $(i_1, j_1), (i_2, j_2) \in R_1$, then $i_1 \leq i_2 \Leftrightarrow k_1 \leq k_2$. This is to make sure that if there are two single-stranded nucleotide which interacts with some base pairs, the tertiary interactions do not cross (Observation (6)). Similarly, if $(i_1, j_1), (i_2, j_2) \in R_2$, then $j_1 \leq j_2 \Leftrightarrow k_1 \leq k_2$.

Structural alignment: Let $S[1 \ldots m]$ be a query sequence with known triple helix structure $H = (M, P)$, and $T[1 \ldots n]$ be a target sequence with unknown structure. S and T are both sequences of {A, C, G, U}. A structural alignment between S and T is a pair of sequences $S'[1 \ldots r]$ and $T'[1 \ldots r]$ 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 triple helix structural similarity between S' and T', is defined as follows. Let $\eta(i)$ be the corresponding position in S such that $S[\eta(i)] = S'[i]$ according to the position i in S', $\gamma(t_1, t_2)$ be similarity score between two characters t_1 and t_2, $\delta(x_1, y_1, x_2, y_2)$ be similarity score between two base pairs (x_1, y_1) and (x_2, y_2), and $\phi(x_1, y_1, z_1, x_2, y_2, z_2)$ be similarity score between two tertiary interactions $(x_1, y_1) * z_1$ and $(x_2, y_2) * z_2$, where $t_1, t_2 \in \{A, C, G, U, \text{'\_'}\}$ and $x_1, x_2, y_1, y_2, z_1, z_2 \in \{A, C, G, U\}$.

$$
\begin{aligned}
score = & \sum_{i=1}^{r} \gamma(S'[i], T'[i]) + \sum_{\substack{i, j \text{s.t.} (\eta(i), \eta(j)) \in M, \\ S'[i], S'[j], T'[i], T'[j] \neq \text{'\_'}}} \delta(S'[i], S'[j], T'[i], T'[j]) \\
& + \sum_{\substack{i, j, k \text{s.t.} (\eta(i), \eta(j)) * \eta(k) \in P, \\ S'[i], S'[j], S'[k], T'[i], T'[j], T'[k] \neq \text{'\_'}}} \phi(S'[i], S'[j], S'[k], T'[i], T'[j], T'[k])
\end{aligned}
$$

There are three parts in the equation. The first part is the sequence similarity score between the two sequences. The second part is the similarity score of the base pairs which are not involved in tertiary interaction. The third part is the similarity score of the tertiary interactions. The problem is to find an alignment to maximize the score. Higher score represents higher similarity between the two sequences according to their sequences and structures. Also, if the score is high, then the alignment can reasonably reveal the triple helix structure of the target sequence.

3. ALGORITHM FOR STANDARD TRIPLE HELIX

In this section, we provide the details of our alignment algorithm for standard triple helix.

Substructure: We solve the problem using dynamic programming. Let $S[1 \ldots m]$ be a query sequence with known standard triple helix structure $H = (M, T)$. Note that x_1 and x_2 are known. Let $v = (a, b, c)$ be a triple with $1 \leq a < x_1 \leq b < x_2 \leq c \leq m$. Similar to Han et al. (2008), we define a subregion based on three points on the sequence (Fig. 2b in which the subregion is highlighted in bold). The *subregion* $R(S, v)$ is defined as $[a, b] \cup [c, m]$. However, because a single-stranded base may interact with more than one base pair and a base pair may interact with more than one single-stranded base, we have to add some rules to define in which circumstance the subregion is valid. $R(S, v)$ is *valid* if it complies with the following conditions:

- All base pairs are either with both end points inside or outside the subregion, i.e., for any $(i, j) \in M$, $i \in R(S, v) \Leftrightarrow j \in R(S, v)$
- If there exists tertiary interaction such that the base pair is inside the subregion but the single-stranded base is not, then the base pair is at the end point of the subregion (Fig. 2c), i.e., for any $(i, j) * k \in P$, if $i, j \in R(S, v)$ and $k \notin R(S, v)$, then $i = a$ and $j = b$, or $i = b$ and $j = c$.
- Similarly, if there exists tertiary interaction such that the single-stranded base is inside the subregion but the base pair is not, then the single-stranded base is at the end point of the subregion (Fig. 2d), i.e., for any $(i, j) * k \in P$, if $k \in R(S, v)$ and $i, j \notin R(S, v)$, then $k = a$ or $k = c$.

Given a valid subregion R in S where $R = R(S, v)$ and $v = (a, b, c)$, in order to solve the problem by dynamic programming, we let the maximum valid subregion inside but smaller than R be $\hat{R} = R(S, v')$. There are five possible cases of which at least one must be satisfied: (I) $v' = (a, b, c + 1)$; (II) $v' = (a + 1, b, c)$; (III) $v' = (a, b - 1, c)$; (IV) $v' = (a + 1, b - 1, c)$; and (V) $v' = (a, b - 1, c + 1)$. If $\hat{R}$ is a valid subregion in more than one case, we set $\hat{R}$ be the case with the smallest case number. The following lemmas prove that at least one of the cases must be satisfied.

Lemma 1. *Let $v = (a, b, c)$. Given that R is valid, then if none of a, b, or c is a single base, then either (a, b) or (b, c) is a base pair.*

Proof by contradiction: Assume that (a, b) and (b, c) are not base pairs. Since a and c are not single base, let (a, b'), (b'', c) be base pairs where $b', b'' < b$ (if $b', b'' > b$, then R is not valid). Since b is not a single base, b should form a base pair with $a' \neq a$ or $c' \neq c$ which however will make R invalid (Fig. 3a). ■

Lemma 2. *If none of a, b, c is a single base and say (a, b) is a base pair, then $\hat{R}$ is valid for $v' = (a + 1, b - 1, c)$.*

Proof. If c is not a single base, there exists a base pair (b', c) where $b' < b$. As shown in Figure 3b, since there cannot exist $c' > c$ that interacts with (a, b) (otherwise R would not be valid), $\hat{R}$ is valid. ■

The following lemmas focus on the cases that one (say a) of a, b, c is a single base. Lemmas 3 and 4 consider the cases when only a is a single base. Lemma 5 considers the case when a, b are single bases and c can be a single base or not. Lemma 6 considers the case when a, c are single bases but b is not. Note that the situation is similar when c is a single base. If b is a single base, then one can refer to Lemma 5.

Lemma 3. *If a is a single base and (b, c) is a base pair, then $\hat{R}$ is valid when v' either is $(a + 1, b, c)$ or $(a, b - 1, c + 1)$.*

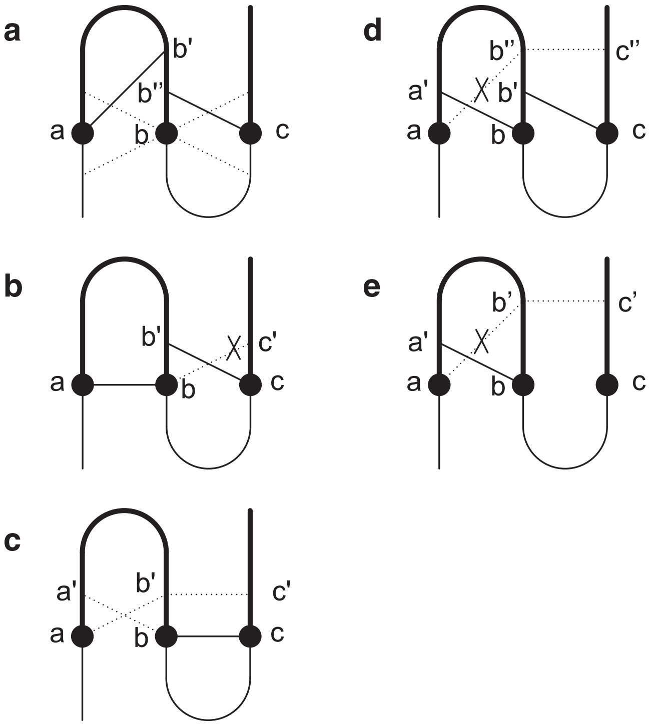


FIG. 3. Region indicated by think lines are subregion $R = R(S, v)$ where $v = (a, b, c)$. Assume the subregion R is valid. **(a)** Explanation for lemma 1: if base pairs (a, b') and (b'', c) exist, then b cannot form a base pair with $a' \neq a$ or $c' \neq c$. Otherwise, R will be invalid. **(b)** Explanation for lemma 2: if base pairs (a, b) and (b', c) exist, then there cannot exist $c' > c$ such that it interacts with the base pair (a, b). Otherwise, R will be invalid. **(c)** Explanation for lemma 3: if base pair (b, c) exists and a is a single base, then only (I) a interacts with a base pair (b', c') where $b' < b$ and $c' > c$; or only (II) $a' > a$ interacts with the base pair (b, c); (but not both) can happen. **(d)** Explanation for lemma 4: if base pairs (a', b) and (b', c) where $a' > a$ and $b' < b$ exist, then there cannot exist a base pair (b'', c'') where $b'' < b$ and $c'' \geq c$ such that it interacts with a. **(e)** Explanation for lemma 6. If base pair (a', b) where $a' > a$ exists, then there cannot exist a base pair (b', c') where $b' < b$ and $c' > c$ such that it interacts with a.

Proof. Since the conditions that (1) a interacts with (b', c') where $b' < b$ and $c' > c$ and (2) (b, c) interacts with a' where $a' > a$ cannot occur together (Fig. 3c), $\hat{R}$ is valid when v' is $(a + 1, b, c)$ or $(a, b - 1, c + 1)$. ■

Lemma 4. *If a is a single base but both b and c are not single base and* (b, c) *is not a base pair, then* $\hat{R}$ *is valid when* $v' = (a + 1, b, c)$.

Proof. Since c is not a single base and (b, c) is not a base pair, there exists a base pair (b', c) where $b' < b$. Also there exists a base pair (a', b) where $a' > a$ because the base pair (b, c') where $c' > c$ does not exist (otherwise R is not valid). Thus a cannot interact with any base pair (b'', c'') where $b'' \leq b$ and $c'' \geq c$ (Fig. 3d). Thus $\hat{R}$ is valid when $v' = (a + 1, b, c)$. ■

Lemma 5. *If a and b both are single bases (c can be a single base or not),* $\hat{R}$ *is valid when* $v' = (a, b - 1, c)$.

Proof. since b is a single base and it cannot interact with any base pair according to the definition of standard helix, $\hat{R}$ is valid when $v' = (a, b - 1, c)$. ■

Lemma 6. *If a and c (but not b) are single bases, then* $\hat{R}$ *is valid when* v' *is* $(a + 1, b, c)$ *or* $(a, b, c + 1)$.

Proof. Since b is not a single base, if a base pair (a', b) where $a' > a$ exists, then $\hat{R}$ is valid when v' is $(a + 1, b, c)$ because a cannot interact with (b', c') where $b' < b$ and $c' > c$ (Fig. 3e). Similarly, if a base pair (b, c') where $c' > c$ exists, then $\hat{R}$ is valid when v' is $(a, b, c + 1)$. ■

The above lemmas consider all situations and the following theorem gives the conclusion.

Theorem 1. *Given a valid subregion R in S where* $R = R(S, v)$ *and* $v = (a, b, c)$*, at least one of the five possible cases (i.e. (I)* $v' = (a, b, c + 1)$*; (II)* $v' = (a + 1, b, c)$*; (III)* $v' = (a, b - 1, c)$*; (IV)* $v' = (a + 1, b - 1, c)$*; and (V)* $v' = (a, b - 1, c + 1)$*) must be satisfied.*

Dynamic programming: Let $S[1, m]$ be the query sequence with triple helix structure $H = (M, P)$ and $T[1, n]$ be the target sequence with unknown structure. Note that x_1 and x_2 of the query sequence S can be chosen appropriately according to the corresponding structure. We can apply the definition of R to T. For any $w = (p, q, r)$ such that $1 \leq p < q < r \leq n$, we define the subregion $R(T, w) = [p, q] \cup [r, n]$. We further define the secondary structure, tertiary interaction and triple helix structure inside a valid *subregion R* as: $Sec(R) = \{(i, j) \in M | i, j \in R\}$, $Tert(R) = \{(i, j) * k \in P | i, j, k \in R\}$ and $Helix(R) = (Sec(R), Tert(R))$. Obviously, $Helix(R)$ is also a standard triple helix structure. Define $B(R, R')$ be the score of the optimal alignment between a subregion R in S with triple helix structure $Helix(R)$ and a subregion R' in T. Note that only the tertiary interaction of which both end points are inside the subregion R would be included in the triple helix structure $Helix(R)$. The score of the optimal alignment between S and T can be obtained by setting $v* = (1, x_2 - 1, x_2)$ which includes the whole query sequence S and the entry $max_{x'_2}\{B(R(S, v^*), R(T, w = (1, x'_2 - 1, x'_2)))\}$, provides the optimal score. Since in the standard triple helix structure, a single-stranded nucleotide may interact with more than one base pair and one base pair may interact with more than one single-stranded nucleotide, we need to consider case by case when computing the value $B(R, R')$.

Let $R = R(S, (a, b, c))$ and $R' = R(T, (p, q, r))$, we also define $C_z(R, R')$ where $z \in \{$'L', 'R', 'LP', 'RP'$\}$ be the optimal alignment score between R and R' with additional requirement: $S[a]$ aligns with $T[p]$ if z = 'L'; $S[c]$ with $T[r]$ if z = 'R'; $(S[a], S[b])$ with $(T[p], T[q])$ if z = 'LP'; $(S[b], S[c])$ with $(T[q], T[r])$ if z = 'RP'; And define $D_z(R, R')$ be the optimal alignment score between R and R' with additional requirement: $S[a]$ aligns with space if z = 'L'; $S[c]$ with space if z = 'R'; $S[a]$ with space and/or $S[b]$ with space if z = 'LP'; $S[b]$ with space and/or $S[c]$ with space if z = 'RP'. Note that $C_{\mathrm{LP}}(R, R')$ and $D_{\mathrm{LP}}(R, R')$ are valid only when $(a, b) \in M$, and $C_{\mathrm{L}}(R, R')$ and $D_{\mathrm{L}}(R, R')$ are valid only when $S[a]$ does not belong to any base pair. Similarly, $C_{\mathrm{RP}}(R, R')$ and $D_{\mathrm{RP}}(R, R')$ are valid only when $(b, c) \in M$, and $C_{\mathrm{R}}(R, R')$ and $D_{\mathrm{R}}(R, R')$ are valid only when $S[c]$ does not belong to any base pair.

The value of $B(R_x, R_y)$ can be computed recursively. Assume that $R(S, (a, b, c + 1))$ is a valid subregion (i.e., Case I), there are three situations to consider. (1) MATCH—aligning base c of S with base r of T; (2) INSERT—insert a space on S; (3) DELETE—delete the base c from S. Lemma 7 summarizes these cases.

Lemma 7. *Given a valid subregion $R = R(S, (a, b, c))$ in S and a subregion $R' = R(T, (p, q, r))$ in T, if $R(S, (a, b, c + 1))$ is a valid subregion (i.e., Case I), then* $B(R, R') = \max\{MATCH, INSERT, DELETE\}$, *where*

$$MATCH = \begin{cases} \mathit{if}\,(a,b) * c \in P, \\ \quad \max\begin{cases} C_{LP}(R(S,(a,b,c+1)), R(T,(p,q,r+1))) + \gamma(S[c], T[r]) \\ \ + \phi(S[a], S[b], S[c], T[p], T[q], T[r]) \\ D_{LP}(R(S,(a,b,c+1)), R(T,(p,q,r+1))) + \gamma(S[c], T[r]) \end{cases} \\ \mathit{else}, \\ B(R(S,(a,b,c+1)), R(T,(p,q,r+1))) + \gamma(S[c], T[r]) \end{cases}$$

$$DELETE = B(R(S,(a,b,c+1)), R(T,(p,q,r))) + \gamma(S[c], \text{‘\_’})$$

$$INSERT = \max\begin{cases} //T[p]\ \mathit{aligns\ with\ space} \\ B(R(S,(a,b,c)), R(T,(p+1,q,r))) + \gamma(T[p], \text{‘\_’}) \\ //T[q]\ \mathit{aligns\ with\ space} \\ B(R(S,(a,b,c)), R(T,(p,q-1,r))) + \gamma(T[q], \text{‘\_’}) \\ //T[r]\ \mathit{aligns\ with\ space} \\ B(R(S,(a,b,c)), R(T,(p,q,r+1))) + \gamma(T[r], \text{‘\_’}) \end{cases}$$

For the MATCH case, if (a, b) is a base pair and interacts with c, then there are two situations: (1) (a, b) of S aligns with (p, q) of T. Then $B(R_x, R_y)$ is the sum of $C_{LP}(R(S, (a, b, c + 1)), R(T, (p, q, r + 1)))$, the sequence score between $S[c]$ and $T[r]$, and the score of tertiary interaction between $(a, b) * c$ of S and $(p, q) * r$ of T; (2) $S[a]$ or $S[b]$ (or both) aligns with space. Then $B(R_x, R_y)$ is the sum of $D_{LP}(R(S, (a, b, c + 1)), R(T, (p, q, r + 1)))$ and the sequence score between $S[c]$ and $T[r]$ score. If (a, b) is not a base pair or (a, b) does not interact with c, then $B(R_x, R_y)$ is the sum of $B(R(S, (a, b, c + 1)), R(T, (p, q, r + 1)))$ and the sequence score between $S[c]$ and $T[r]$. For the INSERT case, if $T[p]$ aligns with a space, then $B(R_x, R_y)$ is the sum of $B(R(S, (a, b, c)), R(T, (p + 1, q, r)))$ and the sequence score between $T[p]$ and space. The situation is similar when $T[q]$ or $T[r]$ aligns with a space. For the DELETE case, $B(R_x, R_y)$ is the sum of $B(R(S, (a, b, c + 1)), R(T, (p, q, r)))$ and the sequence score between $S[c]$ and space.

The situation when $R(S, (a + 1, b, c))$ is valid (i.e., Case II) and $R(S, (a + 1, b, c))$ is valid (i.e., Case III) are similar. The following lemma shows how to compute $B(R_x, R_y)$ when $R(S, (a + 1, b - 1, c))$ is valid (i.e., Case IV).

Lemma 8. *Given a valid subregion $R_x = R(S, (a, b, c))$ in S and a subregion $R_y = R(T, (p, q, r))$ in T, if $R(S, (a + 1, b - 1, c))$ is a valid subregion (i.e., Case IV), then* $B(R_x, R_y) = \max\{MATCH, INSERT, DELETE\}$, *where*

$$MATCH = \begin{cases} \mathit{if}\,(a,b) * c \in P, \\ \quad \max\begin{cases} C_R(R(S,(a+1,b-1,c)), R(T,(p+1,q-1,r))) + \gamma(S[a], T[p]) \\ \ + \gamma(S[b], T[q]) + \delta(S[a], S[b], T[p], T[q]) \\ \ + \phi(S[a], S[b], S[c], T[p], T[q], T[r]) \\ D_R(R(S,(a+1,b-1,c)), R(T,(p+1,q-1,r))) + \gamma(S[a], T[p]) \\ \ + \gamma(S[b], T[q]) \end{cases} \\ \mathit{else\ if}\ (a,b) \in M, \\ B(R(S,(a+1,b-1,c)), R(T,(p+1,q-1,r))) + \gamma(S[a], T[p]) \\ + \gamma(S[b], T[q]) + \delta(S[a], S[b], T[p], T[q]) \\ \mathit{else}, \\ B(R(S,(a+1,b-1,c)), R(T,(p+1,q-1,r))) + \gamma(S[a], T[p]) \\ + \gamma(S[b], T[q]) \end{cases}$$

$$DELETE = \max \begin{cases} B(R(S,(a+1,b-1,c)),R(T,(p,q-1,r)))+\gamma(S[a],\text{‘\_’}) \\ +\gamma(S[b],T[q]) \\ B(R(S,(a+1,b-1,c)),R(T,(p+1,q,r)))+\gamma(S[a],T[p]) \\ +\gamma(S[b],\text{‘\_’}) \\ B(R(S,(a+1,b-1,c)),R(T,(p,q,r)))+\gamma(S[a],\text{‘\_’}) \\ +\gamma(S[b],\text{‘\_’}) \end{cases}$$

$$INSERT = //\textit{same as the INSERT case in Lemma 7}$$

For the MATCH case, if (a, b) is a base pair and interacts with c, then there are two situations: (1) c of S aligns with r of T. Then $B(R_x, R_y)$ is the sum of (a) $C_R(R(S, (a+1, b-1, c)), R(T, (p+1, q-1, r)))$, (b) the sequence score between $S[a]$ and $T[p]$, and between $S[b]$ and $T[q]$, (c) the score of base pair between (a, b) of S and (p, q) of T, and (d) the score of tertiary interaction between $(a, b) * c$ of S and $(p, q) * r$ of T; (2) $S[c]$ aligns with space. Then $B(R_x, R_y)$ is the sum of $D_R(R(S, (a+1, b-1, c)), R(T, (p+1, q-1, r)))$ and the sequence score between $S[a]$ and $T[p]$ and between $S[b]$ and $T[q]$. If (a, b) is a base pair but does not interact with c, then $B(R_x, R_y)$ is the sum of (a) $B(R(S, (a+1, b-1, c)), R(T, (p+1, q-1, r)))$, (b) the sequence score between $S[a]$ and $T[p]$ and between $S[b]$ and $T[q]$, and (c) the score of base pair between (a, b) of S and (p, q) of T. Finally, if (a, b) is not a base pair, then $B(R_x, R_y)$ is the sum of $B(R(S, (a+1, b-1, c)), R(T, (p+1, q-1, r)))$ and the sequence score between $S[a]$ and $T[p]$ and between $S[b]$ and $T[q]$.

For the INSERT case, it is the same as the INSERT case in Lemma 7.

For the DELETE case, there are three situations: (1) only $S[a]$ aligns with a space. Then $B(R_x, R_y)$ is the sum of $B(R(S, (a+1, b-1, c)), R(T, (p, q-1, r)))$ and the sequence score between $S[a]$ and space, and between $S[b]$ and $T[q]$. (2) only $S[b]$ aligns with a space. Then $B(R_x, R_y)$ is the sum of $B(R(S, (a+1, b-1, c)), R(T, (p+1, q, r)))$ and the sequence score between $S[b]$ and space, and between $S[a]$ and $T[p]$. (3) both $S[a]$ and $S[b]$ align with spaces. Then $B(R_x, R_y)$ is the sum of $B(R(S, (a+1, b-1, c)), R(T, (p, q, r)))$ and the sequence score between $S[a]$ and space, and between $S[b]$ and space.

The situation when $R(S, (a, b-1, c+1))$ is valid (i.e., Case V) are similar. The following lemma shows how to compute $C_L(R, R')$ for Case I.

Lemma 9. *For case I - if $R(S, (a, b, c+1))$ is a valid subregion, then*

$$C_L(R,R') = \max \begin{cases} //\textit{MATCH} - S[c] \textit{ aligns with } T[r] \\ C_L(R(S,(a,b,c+1)),R(T,(p,q,r+1)))+\gamma(S[c],T[r]) \\ //\textit{DELETE} - S[c] \textit{ aligns with space} \\ C_L(R(S,(a,b,c+1)),R(T,(p,q,r)))+\gamma(S[c],\text{‘\_’}) \\ //\textit{INSERT} \\ C_L(R(S,(a,b,c)),R(T,(p,q-1,r)))+\gamma(T[q],\text{‘\_’}) \\ C_L(R(S,(a,b,c)),R(T,(p,q,r+1)))+\gamma(T[r],\text{‘\_’}) \end{cases}$$

For the MATCH case, base c of S aligns with base r of T. Then $C_L(R, R')$ is the sum of $C_L(R(S, (a, b, c+1)), R(T, (p, q, r+1)))$ and the sequence score between $S[c]$ and $T[r]$. For the INSERT case, a space is inserted on S. Since $C_L(R, R')$ requires base a of S aligns with base p of T, we consider two situations: (1) $T[q]$ aligns with a space; and (2) $T[r]$ aligns with a space. If $T[q]$ aligns with a space, then $C_L(R, R')$ is the sum of $C_L(R(S, (a, b, c)), R(T, (p, q-1, r)))$ and the sequence score between $T[q]$ and space. The situation is similar when $T[r]$ aligns with a space. For the DELETE case, the base c is deleted from S. $C_L(R, R')$ is the sum of $C_L(R(S, (a, b, c+1)), R(T, (p, q, r)))$ and the sequence score between $S[c]$ and space.

The lemmas for other cases (i.e., Cases II, III, IV, and V) and the calculations of C_R, D_L and D_R are similar.The following lemma shows how to compute $D_{LP}(R, R')$ for Case I.

Lemma 10. *For case I - if $R(S, (a, b, c+1))$ is a valid subregion, then*

$$D_{LP}(R_x, R_y) = \max \begin{cases} //MATCH - S[c] \text{ aligns with } T[r] \\ D_{LP}(R(S, (a, b, c+1)), R(T, (p, q, r+1))) + \gamma(S[c], T[r]) \\ //DELETE - S[c] \text{ aligns with space} \\ D_{LP}(R(S, (a, b, c+1)), R(T, (p, q, r))) + \gamma(S[c], \text{`\_'}) \\ //INSERT \\ D_{LP}(R(S, (a, b, c)), R(T, (p+1, q, r))) + \gamma(T[p], \text{`\_'}) \\ D_{LP}(R(S, (a, b, c)), R(T, (p, q-1, r))) + \gamma(T[q], \text{`\_'}) \\ D_{LP}(R(S, (a, b, c)), R(T, (p, q, r+1))) + \gamma(T[r], \text{`\_'}) \end{cases}$$

$D_{\mathrm{LP}}(R, R')$ requires $S[a]$ or $S[b]$ (or both) to align space. For the MATCH case, $D_{\mathrm{LP}}(R_x, R_y)$ is the sum of $D_{\mathrm{LP}}(R(S, (a, b, c+1)), R(T, (p, q, r+1)))$ (which also requires $S[a]$ or $S[b]$ or both to align space) and the sequence score between $S[c]$ and $T[r]$. For the INSERT case, since it requires $S[a]$ or $S[b]$ (or both) to align space, when $T[p]$ aligns with space, $D_{\mathrm{LP}}(R, R')$ is the sum of $D_{\mathrm{LP}}(R(S, (a, b, c)), R(T, (p+1, q, r)))$ and the sequence score between $T[p]$ and space. The situation is similar for $T[q]$ aligning with space and $T[r]$ aligning with space. For the DELETE case, $D_{\mathrm{LP}}(R, R')$ is the sum of $D_{\mathrm{LP}}(R(S, (a, b, c+1)), R(T, (p, q, r)))$ and the sequence score between $S[c]$ and space.

The lemmas for other cases (i.e., Cases II, III, IV, and V) and the calculations of C_{LP}, C_{RP} and D_{RP} are similar.

To fill the dynamic programming table, not all recursive decompositions of S need to be filled. For a given triple $v = (a, b, c)$ such that $R(S, v)$ is a valid subregion, we can define a function $\zeta(v)$ to determine for which subregions in S, we need to fill the corresponding B, C, and D entires.

$$\zeta(v) = \begin{cases} (a, b, c+1), & \text{if } R(S, (a, b, c+1)) \text{ is valid} \\ (a+1, b, c), & \text{else if } R(S, (a+1, b, c)) \text{ is valid} \\ (a, b-1, c), & \text{else if } R(S, (a, b-1, c)) \text{ is valid} \\ (a+1, b-1, c), & \text{else if } R(S, (a+1, b-1, c)) \text{ is valid} \\ (a, b-1, c+1), & \text{else} \end{cases}$$

Let $v^* = (1, x_2 - 1, x_2)$. We only need to fill in the entries for B, C, and D provided that v^* can be obtained from v by applying ζ function repeatedly. 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 11. *For any sequence $S[\ldots m]$ with standard triple helix structure and any sequence $T[1\ldots n]$, the optimal structural alignment score between $S[1\ldots m]$ and $T[1\ldots n]$ can be computed in $O(mn^3)$.*

4. EXPERIMENTAL RESULTS

An important application is to identify the ncRNAs of the same family with standard triple helix structure along the genome. By inputting a query ncRNA sequence (Q) and its secondary structure, the program can scan a long DNA sequence (T) and output the score for every region in T. A higher score indicates that the sequence and the structure of the region are more similar to those of Q. We performed the experiment as follows: we selected three ncRNA families—RF00024, RF01050, and RF01074—from the Rfam 9.1 database. These families contain a triple helix inside the structure. The corresponding common triple helix structure of each family can be deduced from Chastain and Tinoco (1992), Chen and Greider (2005), Qiao and Cech (2008), and Su et al. (1999). For each family, we extracted the triple helix region of one of the seed members (in the Rfam 9.1 database, for each family, there is a set of reliable members which are regarded as seed members) as a query sequence. To demonstrate the power of structural alignment, the triple helix region selected has the lowest *sequence* similarity with the triple helix region of the other members. Then we created several long random sequences with different percentages of GC content to simulate different regions in a real genome, and we embedded all the whole ncRNA sequences (seed members or non-seed members) of the family (including the sequence of which the triple helix region

Table 1. Summary of Our Results (Same for 50%/75% GC Content)

Family	*No. of real hits*	*No. of identified*	*Sensitivity*
RF00024	117	113	96.6%
RF01050	52	52	100%
RF01074	10	10	100%

has been chosen as query sequence) into this long random sequence in arbitrary positions. The resulting sequence is our *T*. For every region in *T* with length similar to that of the query sequence,[1] we compute the structural alignment score of the region and the query sequence. The details of the families including the sequence selected as the query, the length of the triple helix region of the sequence, and the number of members in each family are given in Table 2.

We assume that regions other than the triple helix region of the real members of the family are false hits as they are likely either not to be members of the family or not the helix regions of the members. Figure 4 shows the distribution of the alignment scores of the true hits (real members) and false hits for all the three families. To compute the effectiveness of our method, we set a threshold as the maximum score of the false hits. We assume that the method finds a real hit if the score of the region is larger than this threshold. Thus a real hit will be missed if the computed score is smaller than or equal to this threshold. We also try different thresholds and the results are similar. Table 1 summarizes the results. Our method can exhibit high sensitivity. For the families RF01050 and RF01074, our method can identify the triple helix region of all the ncRNA sequences along the genome. The sensitivity is 100%. While for the family RF00024, our method can locate 113 out of 117 regions and the sensitivity reaches 96.6%. Figure 4 shows the distribution of the alignment scores of the true hits (the triple helix regions of the real members) and false hits. It is quite clear that the real members can be distinguished from the false hits except that there are four missed out of 117 real hits in the family RF00024. Therefore, the method can reliably locate not only the family members along the genome with varying % of GC content but also the triple helix region from the ncRNA sequence.

There is no existing software available freely for performing structural alignment for triple helix structures. In order to show the effectiveness of using triple helix structures on identifying ncRNAs, we compare our algorithm with two methods: BLAST and PAL (Han et al., 2008). When performing alignment, BLAST only considers the sequence similarity, while PAL considers both sequence and secondary structure similarity, but not the tertiary interactions. Thus we would like to compare the effectiveness of these methods. We use default parameters for BLAST except that the wordsize is set to 7 to increase its sensitivity. For each family, we use the same query sequence and the random sequence *T* as in the above experiment.

Table 3 summarizes the comparison between our result, PAL's result and BLAST's result. Among 117 members of RF00024, we missed four members while PAL and BLAST both missed five members. Among 10 members of RF01074, we did not miss any member, while PAL also did not miss any members for the genomes with 50% GC content but it missed 6 members for 75% GC content, and BLAST missed six members for both genomes with 50% and 75% GC content. It seems that our algorithm performs better than PAL (which considers both sequence and secondary structure but not tertiary interaction) and PAL performs better than BLAST (which only relies on sequence similarity).

Figure 5 shows the comparison between the distribution of our scores and PAL scores. It seems that our algorithm is able to increase the gap between the scores of real hits and the scores of false positives. Figure 6a shows the detailed scores for our method and BLAST for family RF01074 along the genome with 75% GC content. Among 10 members, BLAST missed six of them. However, all the regions of these 10 members got the highest scores if using our algorithm and thus none of them is missed. To take a closer look at the missing cases for BLAST, we found that the missed sequence is usually not similar to the query sequence only based on sequence similarity while the corresponding structure is similar to that of the query sequence. And the sequences that are found by both tools indeed are similar to the query based only on the primary sequence. This provides evidence showing that only considering primary sequence similarity may not be good enough. Figure 6b shows the detailed scores for PAL for family RF01074 along the genome

[1] We set the length of each region equal to the length of the query plus 20.

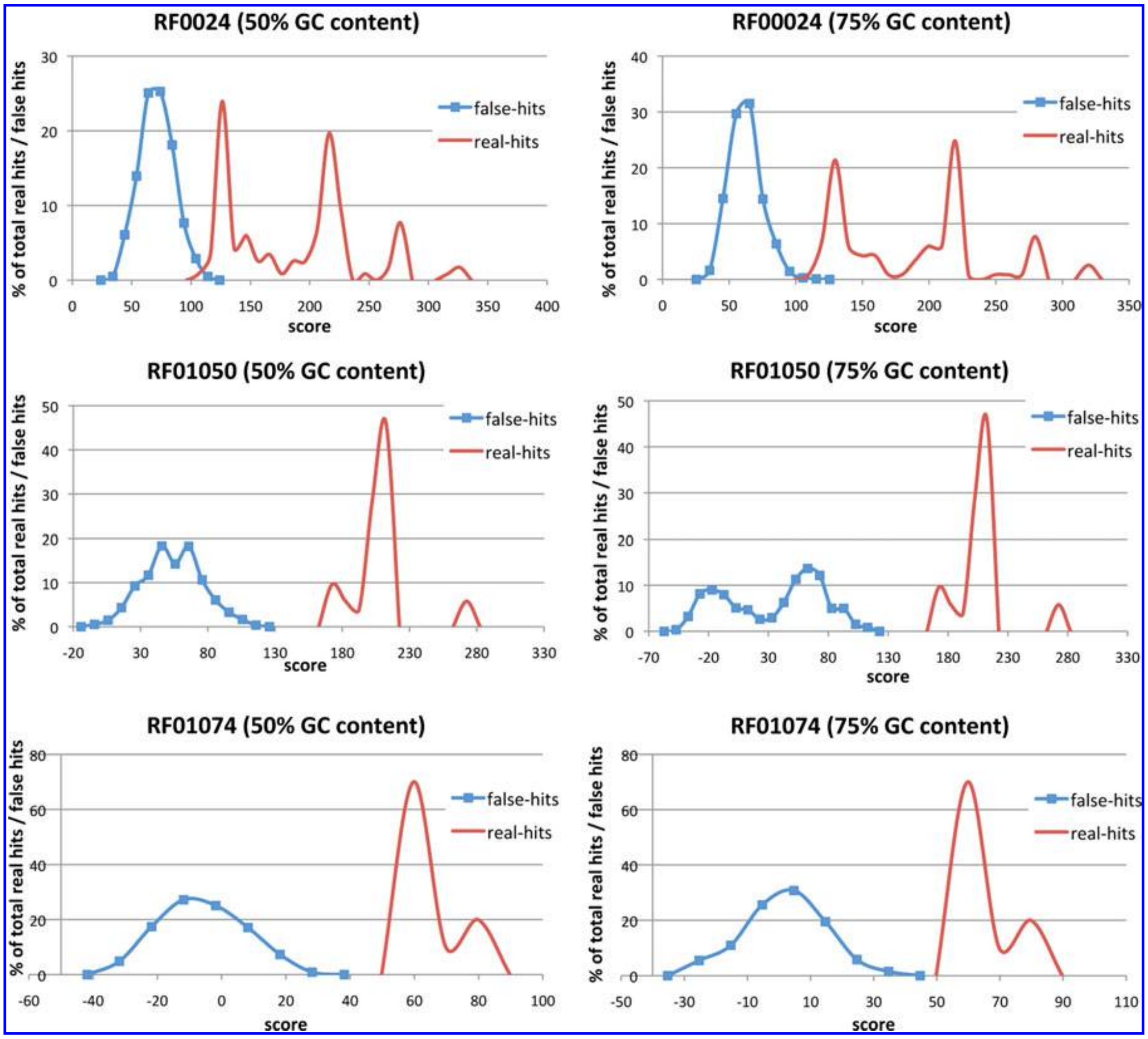


FIG. 4. The distribution of our alignment scores of true hits and false hits for the families RF00024, RF01050, and RF01074 for 50% and 75% GC-content random sequence.

with 75% GC content. There was a false positive with score 50, and only four members with scores higher than 50. Thus, PAL missed six of 10 members. So, considering both the primary sequence and the secondary structure but not the tertiary structure may also miss some of the true hits.

In our experiment, we set the threshold according to the scores of the real hits and the false hits. In the real situation, it may be the case that the true and the false hits are not known in advance. A simple approach is to use a certain percentage of the maximum possible score as a threshold. The maximum possible score is the score when the query sequence is aligned with a sequence exactly the same as the query. According to our experiment, we found that setting the threshold as 35% of the maximum possible score can provide a reasonable result (Table 4). For RF00024, the sensitivity can reach 97%, while the specificity is 99%. For RF01050 and RF01074, the sensitivity can reach 100%, while the specificity is at

Table 2. Details of the ncRNA Families Used in the Experiments

Family	*Query sequence ID*	*Length of triple-helix region*	*Number of members*
RF00024	AF221916.1/94–481	120	117
RF01050	AY639011.1/1–1215	103	52
RF01074	AF352024.1/1581–1620	27	10

Table 3. Comparison Between Our Result, Han's Result, and BLAST Result

			Our result		*Han's result*		*BLAST result*	
Family	*GC content*	*No. of real hits*	*No. of misses*		*No. of misses*		*No. of misses*	
RF00024	50%	117	4	3.4%	5	4.3%	5	4.3%
RF00024	75%	117	4	3.4%	5	4.3%	5	4.3%
RF01050	50%	52	0	0%	0	0%	0	0%
RF01050	75%	52	0	0%	0	0%	0	0%
RF01074	50%	10	0	0%	0	0%	6	60.0%
RF01074	75%	10	0	0%	6	60.0%	6	60.0%

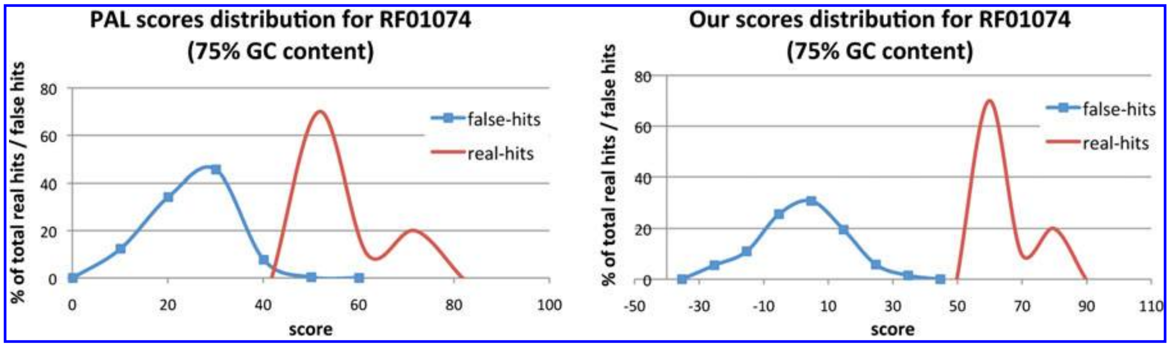


FIG. 5. Comparison the distribution between PAL scores and our scores for the families RF01074 for 75% GC-content random sequence.

least 90%. A more in-depth study should be carried out to derive a better method to set the threshold (e.g., use a similar method based on e-value as suggested by Klein and Eddy [2003]).

Figure 7 shows an example of the deduced triple helix structure for a target sequence. Given a query sequence with standard triple helix structure from the family RF01074 and a target sequence (which is a member in RF01074), according to the resulting alignment between the query and the target outputted by

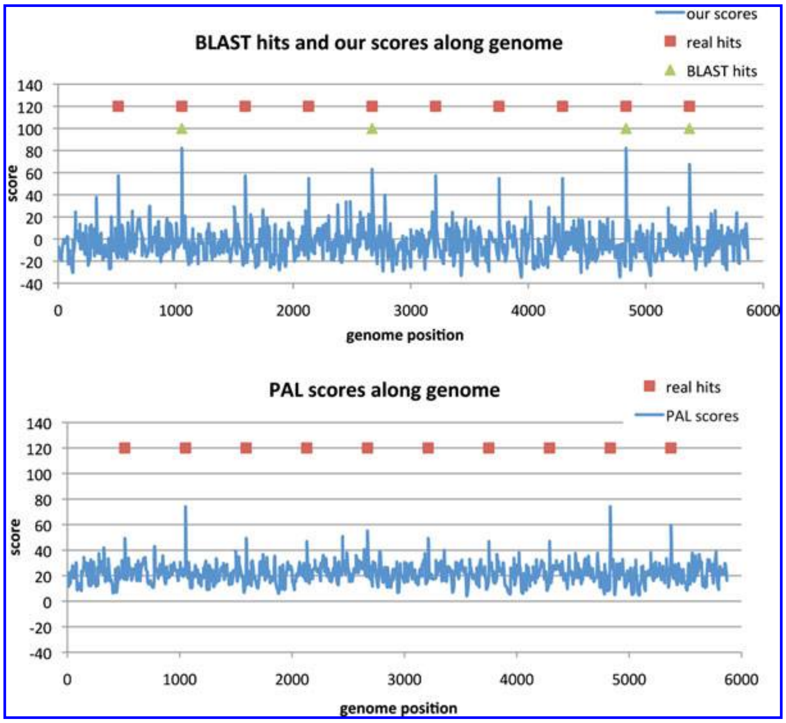
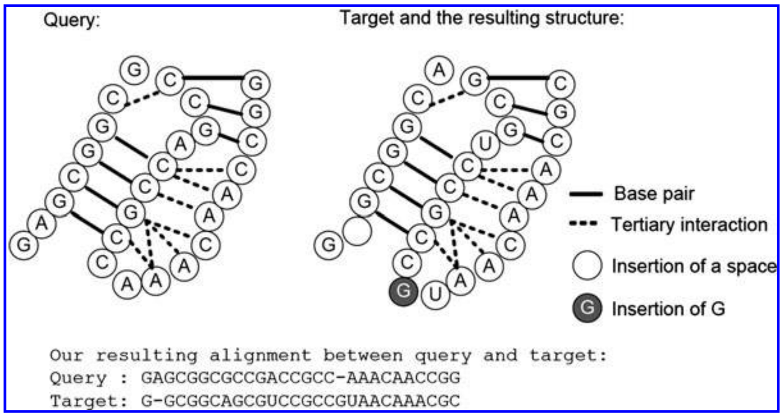


FIG. 6. **(a)** The distribution of our scores and BLAST hits along genome (with 75% GC content) for the family RF01074. Note that BLAST misses six of 10 members. For our scores, the maximum score of the false positives is 40, and all of the scores of 10 members were higher than 40. Thus, our method did not miss any members. **(b)** The distribution of PAL scores along genome (with 75% GC content) for the family RF01074. Note that there was a false positive with score 50, and only four members had scores higher than 50. Thus, Han's method missed six of 10 members.

Table 4. Our Result When Setting a Threshold as 35% of the Scores of All Matches (i.e., the Score When Aligning with a Sequence Exactly the Same as the Query)

Family	*GC content*	*No. of real hits*	*No. of misses*	*Sensitivity*	*No. of false positive*	*Specificity*
RF00024	50%	117	4	97%	1	99%
RF00024	75%	117	4	97%	1	99%
RF01050	50%	52	0	100%	13	80%
RF01050	75%	52	0	100%	13	80%
RF01074	50%	10	0	100%	0	100%
RF01074	75%	10	0	100%	1	91%



```
Our resulting alignment between query and target:
Query : GAGCGGCGCCGACCGCC-AAACAACCGG
Target: G-GCGGCAGCGUCCGCCGUAACAAACGC
```

FIG. 7. An example of the resulting triple helix structure for a target sequence. **(Left)** Query sequence with standard triple helix structure from the family RF01074. **(Right)** According to our resulting alignment between the query and the target, the triple helix structure of the target can be deduced as shown. This target is in fact a member in the family RF01074 and the deduced structure is consistent with the structure stated in Rfam.

our method, the triple helix structure of the target can be deduced (Fig. 7), and the resulting structure is consistent with the structure stated in Rfam.

Regarding the running time (our machine has 8G memory and a dual-core 2.6-GHz CPU), for the query and target of length 120, PAL requires around 60 seconds, our method needs around 70 seconds, and BLAST needs only 1 second. BLAST runs the fastest while both our method and PAL use similar amount of time. Regarding the scores we use when aligning the tertiary interaction in the experiment, we set 1 mark for two aligned interactions if the corresponding base pairs and single-stranded nucleotides are found to be aligned in any family, otherwise a high penalty (e.g., -5) is given. The reason for the setting is to prevent the breaking of tertiary interaction due to the mutation of the bases. Further tuning on the scores should be carried out once we have a better understanding on the tertiary interactions. For base pair alignment, we use the same scoring scheme as in Klein and Eddy (2003).

5. CONCLUSION

In this article, we provided the first algorithm[2] to handle structural alignment of RNA with standard triple helix structure and show that it is useful for detecting ncRNAs. Although there are only a few families in existing databases that have the information of triple helix structures, we expect that there will be more and more ncRNAs which contain this kind of structures. Thus, it is important to study these structures in details. Further directions include speeding up these algorithms, fine-tuning the model of triple helix structure, and considering other more complicated tertiary structures.

[2] We are informed that a new tool called TRFolder, which tries to solve a similar problem as in this paper, will be released soon. A detailed study on TRFolder as well as a comparison between our software and TRFolder will be needed in the future.

ACKNOWLEDGMENTS

This research was supported in part by the General Research Fund (GRF) of the Hong Kong Government (HKU 719611E).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Chastain, M. and Tinoco, I.J. 1992. A base-triple structural domain in RNA. *Biochemistry* 31, 12733–12741.

Chen, J.-L., and Greider, C.W. 2005. Functional analysis of the pseudoknot structure in human telomerase RNA. *Proc. Natl. Acad. Sci. USA* 102, 8080–8085.

Chen, X., Chamorro, M., Lee, S.I., et al. 1995. Structural and functional studies of retroviral RNA pseudoknots involved in ribosomal frameshifting: nucleotides at the junction of the two stems are important for efficient ribosomal frameshifting. *EMBO J.* 14, 842–852.

Frank, D.N., and Pace, N.R. 1998. Ribonuclease P: unity and diversity in a tRNA processing ribozyme. *Annu. Rev. Biochem*. 67, 153–180.

Han, B., Dost, B., Bafna, V., et al. 2008. Structural alignment of pseudoknotted RNA. *J. Comput. Biol.* 15, 489–504.

Klein, R.J,. and Eddy, S.R. 2003. RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinform.* 4, 44.

Le, S., Chen, J., and Maizel, J. 1990. Efficient searches for unusual folding regions in RNA sequences, 127–130. *In: Structure and Methods: Human Genome Initiative and DNA Recombination. Volume 1.* Adenine Press.

Matsui, H., Sato, K., and Sakakibara, Y. 2005. Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures. *Bioinformatics* 21, 2611–2617.

Nguyen, V.T., Kiss, T., Michels, A.A., et al. 2001. 7SK small nuclear RNA blinds to and inhibits the activity of CDK9/cyclin T complexes. *Nature* 414, 322–325.

Qiao, F., and Cech, T.R. 2008. Triple-helix structure in telomerase RNA contributes to catalysis. *Nat. Struct. Mol. Biol.* 15, 634–640.

Rivas, E., and Eddy, S.R. 2000. Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics* 16, 583–605.

Su, L., Chen, L., Egli, M., et al. 1999. Minor groove RNA triplex in the crystal structure of a ribosomal frameshifting viral pseudoknot. *Nat. Struct. Biol.* 6, 285–292.

Theimer, C.A., Blois, C.A., and Feigon, J. 2005. Structure of the human telomerase RNA pseudoknot reveals conserved tertiary interactions essential for function. *Mol. Cell* 17, 671–682.

Wong, T., Lam, T.W., Sung, W.K., et al. 2009. Structural alignment of RNA with complex pseudoknot structure. *Proc. WABI 2009* 403–414.

Yang, Z., Zhu, Q., Luo, K., et al. 2001. The 7SK small nuclear RNA inhibits the CDK9/cyclin T1 kinase to control transcription. *Nature* 414, 317–322.

Zhang, S., Haas, B., Eskin, E., et al. 2005. Searching genomes for noncoding RNA using FastR. *IEEE/ACM TCBB* 2, 4.

Address correspondence to:
Dr. Thomas K.F. Wong
Department of Computer Science
The University of Hong Kong
Hong Kong

E-mail: kfwong@cs.hku.hk