Structural Alignment of Pseudoknotted RNA

Banu Dost*, Buhm Han*, Shaojie Zhang, and Vineet Bafna

Department of Computer Science and Engineering,
University of California, San Diego,
La Jolla, CA 92093-0404, USA
{bdost, buhan, shzhang, vbafna}@cs.ucsd.edu

Abstract. In this paper, we address the problem of discovering novel non-coding RNA (ncRNA) using primary sequence, and secondary structure conservation, focusing on ncRNA families with pseudo-knotted structures. Our main technical result is an efficient algorithm for computing an optimum structural alignment of an RNA sequence against a genomic substring. This algorithm finds two applications. First, by scanning a genome, we can identify novel (homologous) pseudoknotted ncRNA, and second, we can infer the secondary structure of the target aligned sequence. We test an implementation of our algorithm (PAL), and show that it has near-perfect behavior for predicting the structure of many known pseudoknots. Additionally, it can detect the true homologs with high sensitivity and specificity in controlled tests. We also use PAL to search entire viral genome and mouse genome for novel homologs of some viral, and eukaryotic pseudoknots respectively. In each case, we have found strong support for novel homologs.

1 Introduction

Ribonucleic acid (RNA) is the third, and (until recently) most underrated of the trio of molecules that govern most cellular processes: the other two being proteins and DNA. While much of cellular RNA carries a message encoding an amino-acid sequence, other, 'non-coding' RNA participate directly in performing essential functions. Recent and unanticipated discoveries of novel ncRNA families [1, 2, 3, 4, 5] point to the possibility of a 'Modern RNA world' in which RNA molecules are as abundant, and diverse as protein molecules [6]. The analog of the computational gene-finding problem: "given genomic DNA, identify all substrings that encode ncRNA" is increasingly relevant, and relatively unexplored. While potentially abundant, RNA signals are weaker than proteins making them harder to identify computationally. Possibly, the strongest clue is from secondary structure. Being single-stranded, the base-pairs stabilize by forming hydrogen bonds, leading to a characteristic secondary and tertiary structure. With a few exceptions, the base-pairs are *non-crossing*, and form a tree-like structure. This recursive structure is the basis for efficient algorithms to predict RNA structure [7, 8]. With this extensive work in structure prediction, it is natural to expect that novel non-coding RNA could be discovered simply by looking for genomic sub-strings that fold

* These authors contributed equally in the research.

A. Apostolico et al. (Eds.): RECOMB 2006, LNBI 3909, pp. 143–158, 2006.

into low-energy structures. Unfortunately, that idea doesn't work. Rivas and Eddy [9] showed that random DNA (usually with high GC-content) can also 'fold' into low-energy configurations, making it unlikely for a purely *de novo* approach to be successful. Therefore, a comparative approach is employed, often typified by the question: "Given a query RNA with known structure, and a genome, identify all genomic substrings that match the query sequence and structure". The query itself can be either a single molecule or a model (covariance model/stochastic context free grammar) of an RNA structure. This approach has been quite successful and single queries as well as covariance based models are routinely used to annotate genomes with ncRNA [10, 11]. Central to these approaches is an algorithm for computing a local alignment between a query structure and a DNA string. The search itself is simply a scan of the genome to obtain all high scoring local alignments.

Here we pose a related question: *Given a query RNA with known structure, allowing for pseudoknots, and a genome, identify all genomic sub-strings that match the query sequence and structure*. Without being precise, pseudoknots are base-pairs that violate the non-crossing rule (See Figure 1). While not as common as other substructures (bulges,loops), they are often critically important to function. Pseudoknotted RNAs are known to be active as ribozymes [12], self-splicing introns [13], and participate in telomerase activity [14]. They have also been shown to alter gene expression by inducing ribosomal frame-shifting in many viruses [15]. However, understanding the extent and importance of these molecules is partially handicapped by the difficulty of identifying them (computationally). The algorithm presented here will facilitate identification.

In order to compute a local structural alignment, we must start with a formal definition of a pseudoknot in Section 2. Many definitions of pseudoknots have been postulated [16, 17, 18, 19, 20], and recent research investigates the power of these definitions in describing real pseudoknots [21]. We start here with Akutsu's formalism (*simple* pseudoknots) [16], which has a clean recursive structure and encompasses a majority of the known cases [21, 22]. We also present algorithms that extend this class of allowed pseudoknots (standard pseudoknots). Section 3 describes the chaining procedure which is key to the alignment algorithm that follows (Section 4). However, the simple pseudoknots usually do not occur independently, but are embedded in regular RNA structures. In Section 5, we extend the algorithm to handle these cases. Other extensions are considered in Section 7. It has been brought to our attention that a recent publication [23] considers the identical problem using the formation of tree adjoining grammars to model pseudoknots. The pseudoknots considered by them are a restricted version of our simple pseudoknots. Futhermore, our alignment combines sequence and structural similarity. A detailed comparison is deferred to the full version of the paper.

The local alignments can be used in two ways. First, they can be used to infer the structure of the aligned substring that is conserved with the query. We show in Section 8.1 that in a majority of the cases, this leads to a perfect prediction of secondary (pseudoknotted) structure. Next, they can be used to predict novel ncRNA in genomic sequences. While our algorithms are computationally intensive, they can be used in combination with database filtering approaches to search large genomic regions. In Section 8.2, we validate our approach on real sequences embedded in random sequence.

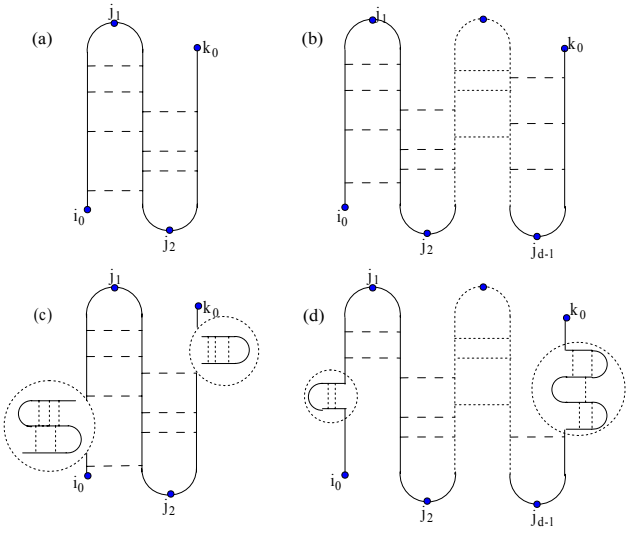


Fig. 1. (a) Simple pseudoknot. (b) Standard pseudoknot of degree d. (c) Recursive simple pseudoknot. (d) Recursive standard pseudoknot of degree d.

Finally, in Section 9, we identify (putative) novel pseudoknotted ncRNA in a search of viral and eukaryotic genomes.

2 Definitions and Preliminary Information

Let $A = a_1...a_n$ be an RNA sequence. The secondary structure is represented simply as the set of base-pairs

$$M = \{(i,j)|1 \leq i < j \leq n, (a_i, a_j) \text{ is a base pair}\}$$

Also, let $M_{i_0,k_0} \subseteq M$ be defined by $M_{i_0,k_0} = \{(i,j) \in M | i_0 \leq i < j \leq k_0\}$. The secondary structure, in the absence of crossing or interweaving base-pairs is called *regular*, and has the following recursive definition.

Definition 1. *An RNA secondary structure M_{i_0,k_0} is regular if and only if $M_{i_0,k_0} = \phi$ or $\exists (i,j) \in M_{i_0,k_0}$ such that*

- $M_{i_0,k_0} = M_{i_0,i-1} \cup M_{i+1,j-1} \cup M_{j+1,k_0} \cup \{(i,j)\}$ *(No base-pairs cross the partitions).*
- *Each of $M_{i_0,i-1}, M_{i+1,j-1}, M_{j+1,k_0}$ is regular.*

Next, we can define the class of allowed pseudoknots ([16]).

Definition 2. *M_{i_0,k_0} is a simple-pseudoknot (see Figure 1(a)) if and only if M_{i_0,k_0} is regular or $\exists j_1, j_2 \in \mathbb{N}$ $(i_0 \leq j_1 < j_2 \leq k_0)$ such that the resulting partition, $D_1 = [i_0, j_1 - 1], D_2 = [j_1, j_2 - 1], D_3 = [j_2, k_0]$, satisfies the following:*

- $M_{i_0,k_0} = (S_L \cup S_R)$, *where* $S_L = \{(i,j) \in M_{i_0,k_0} | i \in D_1, j \in D_2\}$ *and* $S_R = \{(i,j) \in M_{i_0,k_0} | i \in D_2, j \in D_3, \}$.
- S_L *and* S_R *are regular.*

Definition 3. M_{i_0,k_0} *is a standard-pseudoknot with degree* d ($d \geq 3$, *see Figure 1(b)) if and only if* M_{i_0,k_0} *is regular or* $\exists j_1, ..., j_{d-1} \in \mathbb{N}$ $(i_0 \leq j_1 < ... < j_{d-1} \leq k_0)$ *which divide* $[i_0, k_0]$ *into* d *parts,* $D_1 = [i_0, j_1 - 1], D_2 = [j_1, j_2 - 1], ..., D_d = [j_{d-1}, k_0]$, *and satisfy the following:*

- $M_{i_0,k_0} = \bigcup_{l=1}^{d-1} S_l$, *where* $S_l = \{(i,j) \in M_{i_0,k_0} | i \in D_l, j \in D_{l+1}\}$ *for all* $1 \leq l < d$.
- S_l *is regular for all* $1 \leq l < d$,

Note that a simple-pseudoknot is a standard-pseudoknot of degree 3.

Definition 4. M_{i_0,k_0} *is recursive-standard-pseudoknot with degree* d ($d \geq 3$, *see Figure 1(d)) if and only if* M_{i_0,k_0} *is a standard pseudoknot of degree* d *or* $\exists i_1, k_1, ..., i_t, k_t \in \mathbb{N}$ $(i_0 \leq i_1 < k_1 < i_2 < k_2 < ... < i_t < k_t \leq k_0, t \geq 1)$, *which satisfy the following:*

- $(M_{i_0,k_0} - \bigcup_{l=1}^{t} M_{i_l,k_l})$ *is a standard pseudoknot of degree* $\leq d$.
- $M_{i_l,k_l} (1 \leq l \leq t)$ *is a recursive standard pseudoknot of degree* $\leq d$.

A recursive-simple-pseudoknot is a recursive-standard-pseudoknot of degree 3 (Figure 1(c)). While we can devise algorithms to align recursive-standard-pseudoknots, they are computationally expensive, and most known families have a simpler structure. Therefore, we will limit our description and tests to a simpler structure (with a single level of recursion), defined as follows:

Definition 5. M_{i_0,k_0} *is embedded-simple-pseudoknot if and only if* $\exists i_1, k_1, ..., i_t, k_t \in \mathbb{N}$ $(i_0 \leq i_1 < k_1 < i_2 < k_2 < ... < i_t < k_t \leq k_0, t \geq 1)$, *which satisfy the following:*

- $(M_{i_0,k_0} - \bigcup_{l=1}^{t} M_{i_l,k_l})$ *is regular.*
- $M_{i_l,k_l} (1 \leq l \leq t)$ *is a simple-pseudoknot.*

In the full version of the paper, we extend these algorithms to the case of standard-pseudoknots. The full version of the paper will present the algorithm for the most general case (recursive-standard-pseudoknot).

2.1 Structural Alignment Preliminaries

For alignment purposes, we do not distinguish between RNA and DNA, as every substring in the genome might encode an RNA string. Let $q[1 \cdots m]$ and $t[1 \cdots n]$ be two RNA strings over the alphabet $\sum = \{A, C, G, U\}$ where q has a known structure M. An alignment of q and t is defined by a 2-rowd matrix A, in which row 1 (respectively, 2) contains q (respectively, t) interspersed with spaces, and for all columns j, $A[1,j] \neq' -'$ or $A[2,j] \neq' -'$. For $r \in \{1,2\}$, define $\iota_r[i] = i - |\{l < i \text{ s.t. } A[r,l] =' -'\}|$. In other words, if $A[1,i] \neq' -'$, it contains the symbol $q[\iota_1[i]]$. The score of alignment A is given by

$$\sum_j \gamma(A[1,j], A[2,j]) + \sum_{i,j s.t. (\iota_1[i], \iota_1[j]) \in M} \delta(\iota_1[i], \iota_1[j], \iota_2[i], \iota_2[j])$$

The function γ scores for sequence similarity, while δ scores for conservation of structure. While this formulation encodes a linear gap penalty, we note here that alignments of RNA molecules may contain large gaps, particularly in the loop regions, and we implement affine penalties for gaps (details omitted). Naturally, we wish to compute alignments with the maximum score.

The key ideas are as follows: First, note that regular and pseudoknotted structures have a recursive formulation. Therefore, the problem of structurally aligning an RNA structure against a subsequence, can be decomposed into the problems of (recursively) aligning its sub-structures against the appropriate sub-sequences, and combining the results. For regular-structures, the structure is tree-like, and the recursion follows the nodes of the tree. For simple-pseudoknots, the structure is more complex, and will be described in Section 4. The structure for embedded-simple-pseudoknots is simply a combination of the two (See Section 7).

However, it is not sufficient to consider structural elements alone, as we wish to score for sequence conservation as well. The recursive structure described only contains a subset of the nucleotides that participate in structure. Therefore, we employ a second trick of introducing spurious structural elements (base-pairs) to M. The augmented structure M' must have the following properties:

- Each nucleotide i appears in M'.
- $|M'| = O(m)$, so that the size of the structure does not increase too much.
- The recursive structure of M is maintained.

Pseudoknots and regular structures have very different recursive structure, and require different augmentation procedures. In Section 3, we present *chaining*, a novel augmentation procedure for simple pseudoknots. An augmentation for regular structures, *binarization* was presented in [24], and is implicit in the covariance models used to align regular RNA [25]. Here, we extend binarization to include chaining for embedded-simple-pseudoknots (Figure 5). These augmentations are used in the alignment algorithms for simple (Section 4), and embedded-simple-pseudoknots (Section 5).

3 Chaining

Before describing the chaining procedure, we revisit the problem of aligning a simple pseudoknot to a genomic sub-string. Unlike regular structures, we cannot partition the genome into contiguous substrings, because of interweaving base pairs. Thus, we need a new substructure for simple pseudoknot structures.

We start by defining a total ordering among the base pairs of a simple pseudoknot. Recall (Definition 3) that a simple-pseudoknot structure M_{i_0,k_0} can be divided into 3 parts: $D_1 = [i_0, j_0 - 1], D_2 = [j_0, j'_0 - 1], D_3 = [j'_0, k_0]$. (See Figure 2(a)) For each base pair $(i, j) \in M$, exactly one of i and j is in D_2 part. We define an ordering of the base pairs in M by sorting the coordinate in D_2. Formally, define $D_2(i, j)$ for all $(i, j) \in M$ as follows: $D_2(i, j) = i$ if $(i, j) \in S_R$, and $D_2(i, j) = j$ otherwise. For each $(i, j), (i', j') \in M$,

$$(i, j) \geq_p (i', j') \text{ iff } D_2(i, j) \geq D_2(i', j')$$

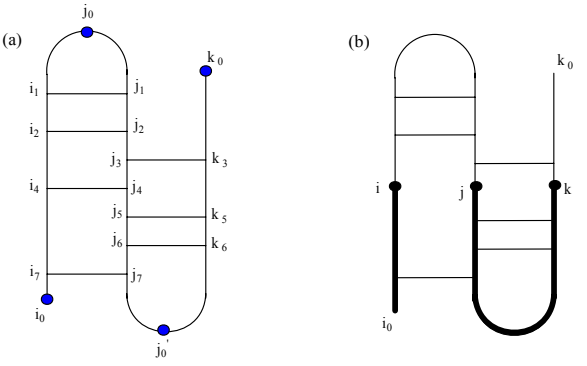


Fig. 2. (a) Base pairs in a simple pseudoknot are ordered according to the index of the endpoint along $[j_0, j_0']$. Therefore, $(i_1, j_1) > (i_2, j_2) > (j_3, k_3) > (i_4, j_4) > (j_5, k_5) > (j_6, k_6) > (i_7, j_7)$. (b) Subpseudoknot structure.

As distinct base-pairs do not share any coordinates, $\geq_p$ defines a total ordering on the actual base-pairs, and can be used to define a partial order on substructures that we can recurse on. Define a *subpseudoknot* $\mathcal{P}(i, j, k)$ as the union of two subintervals $\mathcal{P}(i, j, k) = [i_0, i] \cup [j, k]$ (Figure 2 (b)). Denote the triple (i, j, k) as the *frontier* for $\mathcal{P}(i, j, k)$. Note that i_0 is implicit from the context. Suppose that we are aligning frontier (i', j', k') of the query against frontier (i, j, k) of the target, with the score represented by $B[i, j, k, i', j', k']$. A naive algorithm would need to consider $O(m^3 n^3)$ pairs of frontiers. We improve this as follows: consider the special case of $(i', j') \in M$ where $(i', j') \in S_L$. The following recursion gives the score for B (proof omitted).

Theorem 1

$$B[i, j, k, i', j', k'] = \max\{\textit{MATCH,INSERT,DELETE}\} \tag{1}$$

$$\begin{aligned} \textit{MATCH} = {} & B[i-1, j+1, k, i'-1, j'+1, k'] + \delta(q[i'], q[j'], t[i], t[j]) \\ & + \gamma(q[i'], t[i]) + \gamma(q[j'], t[j]), \end{aligned} \tag{2}$$

$$\textit{DELETE} = \max \begin{cases} B[i-1, j, k, i'-1, j'+1, k'] + \gamma(q[i'], t[i]) + \gamma(q[j'], '-'), \\ B[i, j+1, k, i'-1, j'+1, k'] + \gamma(q[i'], '-') + \gamma(q[j'], t[j]), \\ B[i, j, k, i'-1, j'+1, k'] + \gamma(q[i'], '-') + \gamma(q[j'], '-') \end{cases} \tag{3}$$

$$\textit{INSERT} = \max \begin{cases} B[i-1, j, k, i', j', k'] + \gamma('-', t[i]), \\ B[i, j+1, k, i', j', k'] + \gamma('-', t[j]), \\ B[i, j, k-1, i', j', k'] + \gamma('-', t[k]) \end{cases} \tag{4}$$

Note that in every sub-case of MATCH and DELETE, we move from the query frontier (i', j', k') to the frontier $(i'-1, j'+1, k)$, because if either i' or j' is not used, we cannot score for the pair (i', j'). In the INSERT case, we stay at the frontier (i', j', k'). The situation is symmetric when $(j', k') \in S_R \subseteq M$, but is not defined when $(i', j') \notin M \wedge (j', k') \notin M$. The key idea for the chaining procedure is that we can define a

unique frontier to move to in all cases, and still ensure that each nucleotide is touched by at least one frontier. By starting with a fixed frontier, and always moving to a fixed child, we only have $O(m)$ frontiers to consider.

From Definition 2, there exist indices j_1, j_2 which divide the simple pseudoknot structure into D_1, D_2 and D_3. We choose $(j_1 - 1, j_1, k_0)$ as the *root* frontier. Note that $\mathcal{P}(j_1 - 1, j_1, k_0)$ represents the entire simple-pseudoknot (See Figure 3(a)). We maintain the invariant that if (i, j, k) is a frontier and j participates in a base-pair, then the base-pair must be 'below' or within the frontier. In other words, if $(i', j) \in S_L$, then $i' \leq i$. Likewise, if $(j, k') \in S_R$, then $k' \leq k$. For a frontier (i, j, k), we have different cases: for example, if $(i', j) \in S_L$, we add spurious base pairs $(i, j), (i - 1, j), \ldots (i', j)$. These base pairs define an ordered set of frontiers $(i, j, k) \geq (i - 1, j, k) \geq \ldots, (i', j, k) \geq (i' - 1, j + 1, k)$. Likewise, if $(j, k') \in S_R$, we add spurious base-pairs $(j, k), (j, k - 1), \ldots, (j, k')$, which define the frontiers $(i, j, k) \geq \ldots \geq (i, j + 1, k' - 1)$. The chaining algorithm, with a complete listing of cases is described in Figure 3. The output of chaining is a directed path of 'frontiers'. The number of nucleotides in a frontier (i, j, k) is given by the expression $((i - i_0 + 1) + (k - j + 1)) \leq m$. Further, this number decreases by at least 1 for each adjacent frontier. Thus the

```
CHAINING(i, j, k)
 1  if i = i_0 − 1 and j > k
 2     then return NIL
 3  if (i, j) ∈ S
 4     then v = CHAINING(i − 1, j + 1, k);
 5          return CREATENODE(i, j, solid, move(1, 1, 0), v)
 6  if (j, k) ∈ S
 7     then v = CHAINING(i, j + 1, k − 1);
 8          return CREATENODE(j, k, solid, move(0, 1, 1), v)
 9  if j ∈ V_L
10     then v = CHAINING(i − 1, j, k);
11          return CREATENODE(i, j, empty, move(1, 0, 0), v)
12  if j ∈ V_R
13     then v = CHAINING(i, j, k − 1);
14          return CREATENODE(j, k, empty, move(0, 0, 1), v)
15  if i ∈ V_L
16     then v = CHAINING(i, j + 1, k);
17          return CREATENODE(i, j, empty, move(0, 1, 0), v)
18  if k ∈ V_R
19     then v = CHAINING(i, j + 1, k);
20          return CREATENODE(j, k, empty, move(0, 1, 0), v)
21  if i > i_0
22     then v = CHAINING(i − 1, j, k);
23          return CREATENODE(i, j, empty, move(1, 0, 0), v)
24  if i = i_0
25     then v = CHAINING(i − 1, j + 1, k);
26          return CREATENODE(i, j, empty, move(1, 1, 0), v)
27  if i = i_0 − 1
28     then v = CHAINING(i, j + 1, k);
29          return CREATENODE(j, k, empty, move(0, 1, 0), v)
```

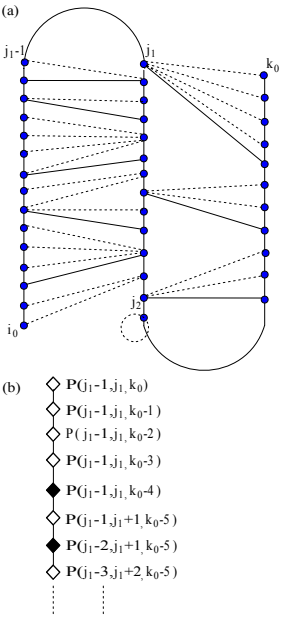


Fig. 3. The chaining procedure on a simple pseudoknot structure M_{i_0,k_0}. (a) Solid base pairs are the actual base pairs, dotted ones are the spurious base pairs. (b) Chain structure representing the simple pseudoknot structure M_{i_0,k_0}. Solid nodes represents a sub-pseudoknot with frontier (i, j, k) where (i, j) or (j, k) is an actual base pair. Empty nodes represents a sub-pseudoknot with frontier (i, j, k) where neither (i, j) nor (j, k) is an actual base pair.

(a) ALIGN-SP($M', t[1...n]$)

1 // M' is the chain representing the simple pseudoknot region to be aligned in query q
2 **for** all intervals (i_0, k_0) in $t[1...n]$
3 **do for** all (i, j, k), $i_0 \leq i < j \leq k \leq k_0$
4 **do for** all nodes $v \in M'$
5 **do if** $v \in M_L$
6 **then** $B[i,j,k,v] = \max \begin{cases} B[i-1, j+1, k, child(v)] + \delta(q[l_v], q[m_v], t[i], t[j]) \\ \qquad +\gamma(q[l_v], t[i]) + \gamma(q[m_v], t[j]), \\ B[i-1, j, k, child(v)] + \gamma(q[l_v], t[i]) + \gamma(q[m_v], '-'), \\ B[i, j+1, k, child(v)] + \gamma(q[l_v], '-') + \gamma(q[m_v], t[j]), \\ B[i, j, k, child(v)] + \gamma(q[l_v], '-') + \gamma(q[m_v], '-') \end{cases}$
7 **if** $v \in M_R$
8 **then** $B[i,j,k,v] = \max \begin{cases} B[i, j+1, k-1, child(v)] + \delta(q[m_v], q[r_v], t[j], t[k]) \\ \qquad +\gamma(q[m_v], t[j]) + \gamma(q[r_v], t[k]), \\ B[i, j, k-1, child(v)] + \gamma(q[m_v], '-') + \gamma(q[r_v], t[k]), \\ B[i, j+1, k, child(v)] + \gamma(q[m_v], t[j]) + \gamma(q[r_v], '-'), \\ B[i, j, k, child(v)] + \gamma(q[m_v], '-') + \gamma(q[r_v], '-') \end{cases}$
9 **if** $v \in M_S$ and $move(v) = (1,0,0)$
10 **then** $B[i,j,k,v] = \max \begin{cases} B[i-1, j, k, child(v)] + \gamma(q[l_v], t[i]), \\ B[i, j, k, child(v)] + \gamma(q[l_v], '-') \end{cases}$
11 **if** $v \in M_S$ and $move(v) = (0,0,1)$
12 **then** $B[i,j,k,v] = \max \begin{cases} B[i, j, k-1, child(v)] + \gamma(q[r_v], t[k]), \\ B[i, j, k, child(v)] + \gamma(q[r_v], '-') \end{cases}$
13 **if** $v \in M_S$ and $move(v) = (0,1,0)$
14 **then** $B[i,j,k,v] = \max \begin{cases} B[i, j+1, k, child(v)] + \gamma(q[m_v], t[k]), \\ B[i, j, k, child(v)] + \gamma(q[m_v], '-') \end{cases}$
15 $B[i,j,k,v] = \max \begin{cases} B[i,j,k,v] \\ B[i-1, j, k, v] + \gamma('-', t[i]), \\ B[i, j+1, k, v] + \gamma('-', t[j]), \\ B[i, j, k-1, v] + \gamma('-', t[k]) \end{cases}$
16
17 $B_{SP}[i_0, k_0, i_{SP}, k_{SP}] = \max_{j=i+1, k=k_0} \{B(i, j, k, \text{ROOT}(M'))\}$

(b) IMPROVED ALIGN-SP()

1 **for** all $v \in M'$
2 **do for** $i_0 = 1$ **to** $n-1$
3 **do for** $i = i_0 - 1$ **to** $n-1$
4 **do for** $j = n+1$ **downto** $i+1$
5 **do for** $k = j-1$ **to** n
6 **do** Compute $B[i,j,k,v]$

Fig. 4. (a) Align-SP procedure for alignment of a simple pseudoknot structure to a target sequence $t[1...n]$. (b) Improved Align-SP procedure.

number of nodes in the chain is $O(m)$. We still need to consider $O(n^3)$ target frontiers in aligning, for a complexity of $O(mn^3)$.

4 Alignment Algorithm for Simple-Pseudoknots

Figure 4(a) describes the algorithm ALIGN-SP for aligning a simple-pseudoknot to a DNA substring. Its input is a chain of query sub-pseudoknots, which is aligned to all sub-pseudoknots $\mathcal{P}(i,j,k)$ of the target sequence $t[1 \ldots n]$. Let M_L (respectively M_R) be the set of solid nodes representing subpseudoknots $\mathcal{P}(i,j,k)$ where $(i,j) \in S_L$ (respectively, $(j,k) \in S_R$). Let M_S be set of the nodes representing subpseudoknots $\mathcal{P}(i,j,k)$ where neither $(i,j) \notin S_L$, and $(j,k) \notin S_R$.

As an example, suppose we are aligning sub-pseudoknot $\mathcal{P}(i,j,k)$ in t to the subchain rooted at v. Let $B[i,j,k,v]$ be the score of the optimal alignment. First, we have

```
BINARIZE-SP(i, j)
 1  if (i, j) is a simple pseudoknot structure
 2    then return CHAINING(i, j, pseudo − node, Nil);
 3  if i = j
 4    then return CREATENODE(i, j, empty, Nil);
 5  if (i, j) ∈ M
 6    then v = BINARIZE-SP(i + 1, j − 1);
 7         return CREATENODE(i, j, solid, v);
 8  if (k, j) ∈ M for some i < k < j
 9    then
10         vl = BINARIZE-SP(i, k − 1);
11         vr = BINARIZE-SP(k, j);
12         (A empty node with 2 children, vl and vr.)
13         return CREATENODE(i, j, empty, vl, vr);
14  if i < j
15    then v = BINARIZE-SP(i, j − 1);
16         return CREATENODE(i, j, empty, v);
```

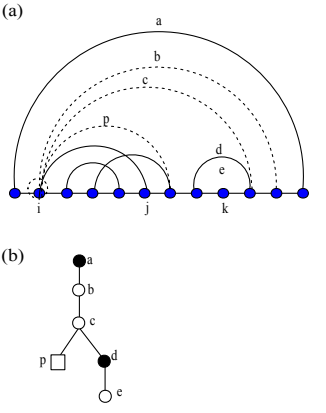


Fig. 5. Binarization procedure revised for embedded-simple-pseudoknots and an illustration. (a) An embedded-simple-pseudoknot with spurious base pairs added. (b) Resulting binary tree. Solid nodes correspond to actual base pairs while empty (circular) nodes correspond to spurious base pairs. A '□' represents a pseudonode and subtree rooted at a pseudonode is formed by Chaining procedure.

cases involving insertion of target nucleotides: $t[i], t[j]$, and $t[k]$, as described by the recurrence in Figure 4(a)(Line 15). Next, we have the cases corresponding to match or deletion of v. We consider the case $v \in M_L$ corresponding to the subpseudoknot $\mathcal{P}(l_v, m_v, r_v)$ in q. The following cases can occur

1. $(t[i], t[j])$ is a pair in t corresponding to the pair $(q[l_v], q[m_v])$ in q.
2. $q[l_v]$ is substituted with $t[i]$ and $q[m_v]$ is deleted.
3. $q[m_v]$ is substituted with $t[j]$ and $q[l_v]$ is deleted.
4. $q[l_v]$ and $q[m_v]$ are both deleted.

The corresponding recurrences are shown on Line 6 of the procedure. The other cases are handled in an analogous fashion and are described in Figure 4.

5 Alignment Algorithm for Embedded-Simple-Pseudoknots

We consider now the special case of aligning recursive-simple-pseudoknots in which simple-pseudoknots are embedded in a regular structure. This is by far the most common occurrence of pseudoknots. While it is relatively easy to extend our algorithms to handle the full generality of recursive-pseudoknots, the complexity increase makes the algorithms untractable for real problems. Thus, this special case offers a compromise between generality and practicality.

The first step in the procedure is to *binarize* the query RNA, so that every nucleotide is in a base-pair, and can be represented by a binary tree of size $O(m)$ [24]. The main difference is that we invoke the chaining procedure whenever a simple-pseudoknot is encountered. Thus, in the binary tree, the simple pseudoknot substructure appears as a chain rooted at a *pseudo-node*.

After the binary tree structure M' of query sequence q is created, target sequence t is aligned to this tree. The following procedure ALIGN aligns a given subsequence $(t[i \dots j])$ in target sequence to a subtree of M'. The scores of optimal alignments are stored in matrix A. The entry $A[i, j, v]$ keeps the optimal alignment of the subproblem of aligning a subsequence $(t[i], t[j])$ to the subtree rooted at the node v, in other words to the subinterval $(q[l_v], q[r_v])$ of the query sequence.

6 Complexity

In Align-SP, lines $3 - 15$ runs in $O(n^3)$ time to align all subpseudoknots in target to a node. Those lines are executed for each subinterval (i_0, k_0) in target and for each node in the query tree. Then, time complexity of procedure Align-SP becomes $O(mn^5)$. However, we do not need to compute $O(n^3)$ scores for each subinterval (i_0, k_0). Since k_0 does not appear in the recurrences of Align-SP procedure and $B[i, j, k]$ does not depend on $B[i', j', k']$ such that $k' > k$, $B[i, j, k]$ does not depend on k_0. Thus, it is enough to compute $O(n^3)$ scores for each i_0 as shown in Figure 4(b). Then, total running time of Align-SP is $O(mn^4)$.

In Align procedure in Figure 6, we first call Binarization-SP procedure which runs in $O(m)$ time. We also call Align-SP procedure whenever we encounter with a pseudo-node in the binary tree formed. Let m_p be the length of the pseudoknot regions in $q[1 \cdots m]$, m_1 and m_2 be the number of the nodes with one child and two children in the binary tree of q representing the regions with regular structure. Then, the total running time of Align procedure will be $O(m_p n^4 + m_1 n^2 + m_2 n^3)$. It is useful to note that very often, $m_p, m_2 \in o(m)$, and so the true complexity is better than the worst case complexity. Also, in computing good alignments, we can often bound the gap-lengths.

ALIGN$(q[1 \dots m], t[1 \dots n])$
1 $M' =$ BINARIZE-SP(Q)
2 **for** all intervals (i, j) in t and all nodes v in M
3 **do if** v is NIL
4 **then** $A[i, j, NIL] = \sum_{l=i}^{j} \gamma(t[l], '-')$
5 **if** v is a **pseudo node**
6 **then** $A[i, j, v] =$ **return** ALIGN-SP(i, j, v)
7 **if** $v \in M$
8 **then** $A[i, j, v] = \max \begin{cases} A[i+1, j-1, \text{child}(v)] + \delta(t[i], t[j], q[lv], q[rv]) \\ A[i, j-1, v] + \gamma('-', t[j]) \\ A[i+1, j, v] + \gamma('-', t[i]) \\ A[i+1, j, \text{child}(v) + \gamma(q[lv], t[i]) + \gamma(q[rv], '-') \\ A[i, j-1, \text{child}(v)] + \gamma(q[lv], '-') + \gamma(q[rv], t[j]) \\ A[i, j, v] + \gamma(q[lv], '-') + \gamma(q[rv], '-') \end{cases}$
9 **if** $v \in M' - M$ and v has one child
10 **then** $A[i, j, v] = \max \begin{cases} A[i, j-1, \text{child}(v)] + \gamma(q[rv], t[j]) \\ A[i, j, \text{child}(v)] + \gamma(q[rv], '-') \\ A[i, j-1, v] + \gamma('-', t[j]) \\ A[i+1, j, v] + \gamma('-', t[i]) \end{cases}$
11 **if** $v \in M' - M$ and v has two children
12 **then** $A[i, j, v] = \max_{i \le k \le j} \{A[i, k-1, \text{left\_child}(v)] + A[k, j, \text{right\_child}(v)]\}$

Fig. 6. Alignment Algorithm for aligning an embedded-simple-pseudoknot $q[1...m]$ to a target sequence $t[1...n]$

To take advantage of this, we employ a banding procedure (details not shown). A discussion of scoring matrices and gap penalties is deferred to the full-version of the paper.

7 Alignment Algorithm for Standard Pseudoknots

It is possible to extend the algorithm for aligning a simple pseudoknot to an alignment algorithm for a standard pseudoknot with degree $d > 3$. In the full version of the paper, we present an extension of our algorithm for standard pseudoknot structures with degree 4, and achieve the following result:

Theorem 2 *The optimal alignment for a standard pseudoknot with degree* 4 *can be computed in* $O(mn^4)$ *time which is identical to the degree* 3 *case (simple pseudoknots). In general, standard pseudoknots of degree* $2k-1$ *and* $2k$ *can be aligned in* $O(mn^{2k})$ *time.*

8 Results

A C++ implementation of the algorithm given for simple pseudoknots (PAL) is done. PAL takes an RNA query and target sequence, and returns all high scoring structural local alignments in the target sequence. All tests were performed on a PC (3.4 Ghz, 1 GB RAM) unless otherwise stated. The structure of the target sub-sequence is inferred from the alignment (Ex: Figure 8). In order to assess the performance of PAL, we tested 6 RNA families from Rfam database: UPSK, Antizyme, Parecho CRE, Corona-FSE, Corona-pk3 and IFN-gamma. Each of these families has an embedded-simple-pseudoknot structure. General information about these families are shown in Table 1.

Table 1. 6 Simple Pseudoknotted RNA families. Avg Id stands for the average sequence identity between two seed members, n for the number of seed members, L for the length, L_P for the length of the pseudoknot region and t for the average time PAL takes for the alignment of a pair.

RNA Family	**Rfam Id**	Avg Id	n	L	L_P	$t(sec)$
UPSK	RF00390	92.78%	4	$23-23$	~ 22	0.0
Antizyme	RF00381	83.07%	13	$57-59$	~ 54	12.8
Parecho CRE	RF00499	81.99%	5	$102-115$	~ 33	1.4
Corona-FSE	RF00507	67.44%	18	$79-85$	~ 76	31.5
Corona-pk3	RF00165	69.42%	14	$62-64$	~ 56	19.9
IFN-gamma	RF00259	89.83%	5	$166-169$	~ 113	51.7

8.1 Predicting Structure with PAL

To test structural inference, we select a pair of members from a family as the query and target. PAL is used to align the query to the target. The inferred structure of the target is compared against the annotated structure in the Rfam database. We evaluate the predicted structure by computing *TP* (true positives), *FP* (false positives) and *FN*

(false negatives), defined as follows: *TP* is the number of base pairs in inferred target structure that are correct: *FP* is the number of base pairs in the inferred structure that are not in the true structure, and *FN* is number of base pairs in the true structure that are not inferred. We define *Specificity = TP / (TP + FP)* and *Sensitivity = TP / (TP + FN)*. Good performance is indicated by both being close to 1. Table 2 summarizes the result of testing each pair in the 6 families. As the results show, PAL is a strong predictor of structure, with mean sensitivity and specificity of 0.95. We also investigated the few cases in which the prediction was away from the mean. In most of those cases, the target had stem loops that were longer than the query. As they were not aligned to the query structure, they were not inferred. In practice, we would augment the inferred structure by a local extension of stem loops in both directions. A second source of errors was incorrect annotation in Rfam. Other than these two scenarios, the structure inference was essentially correct.

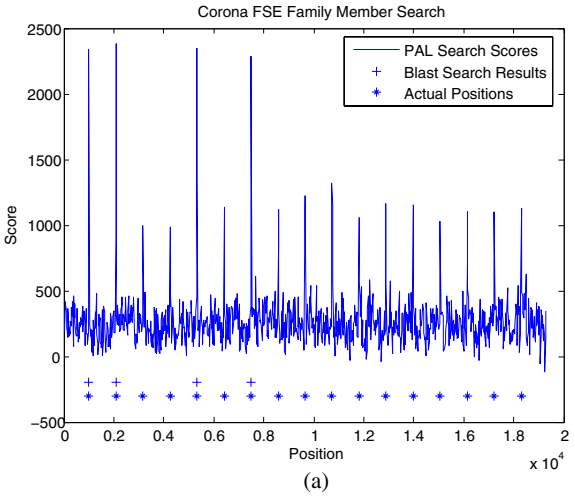
There is a second caveat in these results which is not apparent. Many (but not all) of the sequences have high sequence similarity, which might be making the alignment task easier. We believe this is because a sequence search tool like Blast is used to fish out candidates, which are then manually aligned, and experimentally validated. We will show in the following sections that our tool can pick out candidates that BLAST cannot find, and also align them structurally. Also, in the cases where there isn't high sequence similarity, the structure inference was just as good.

Table 2. Pairwise tests: Statistics for Specificity and Sensitivity values. Mean is the average of Specificity (Sensitivity) values and median is the mid-point of Specificity (Sensitivity) values over all seed member pairs in an RNA family.

	Specificity				**Sensitivity**			
RNA Family	Mean	StdDev	Median	Range	Mean	StdDev	Median	Range
UPSK	1.000	0.000	1.000	(1.000-1.000)	1.000	0.000	1.000	(1.000-1.000)
Antizyme	0.991	0.020	1.000	(0.941-1.000)	0.991	0.020	0.941	(0.941-1.000)
Parecho	0.951	0.052	0.976	(0.848-1.000)	0.938	0.053	0.952	(0.844-1.000)
Corona-FSE	0.944	0.100	1.000	(0.737-1.000)	0.937	0.105	1.000	(0.737-1.000)
Corona-pk3	0.971	0.053	1.000	(0.765-1.000)	0.968	0.056	1.000	(0.722-1.000)
IFN-gamma	0.937	0.092	1.000	(0.782-1.000)	0.934	0.093	1.000	(0.782-1.000)

8.2 Searching for Structural Homologs

In this test, we use one of the members of an RNA family as a query, and look for its homolog in a large random sequence, with the other members inserted. Figure 7(a) shows the results for the Corona-FSE family, in which 17 members were embedded in a 19kb random sequence. The windowed scores are shown by solid lines. The actual positions of the remaining 17 members are denoted by '*'. We note that the true hits are easily the highest scoring regions along the sequence, and that all true positives score higher than all the false hits. The lowest scoring TP has a score of 988 and the highest scoring FP has a score of 606. Moreover, the random sequence scores do not



(a)

RNA Family	# Found BLAST	# Found PAL
UPSK	3	3
Antizyme	12	12
Parecho CRE	4	4
Corona-FSE	4	17
Corona-pk3	5	13
IFN-gamma	4	4

(b)

Fig. 7. Use of PAL as a pseudoknot RNA search tool (a) Score plot for Corona-FSE homologue search. '*' denotes actual positions of the members and '+' denotes the members located by Blastn. (b) Comparison against BLAST on other families.

show a large variation. We do not compute P-values on the hits, but in future work, we will use the distribution of scores on random, or genomic sequence (with differing GC-content) to compute the P-value. In general, the distribution is not understood, and we will either use a non-parametric value such as the Chebyshev's inequality [26], or perhaps the Gumbel distribution, which has been shown to be a good approximation to the actual distribution [11]. In contrast, Blastn (E-value 10, Word-size 7) is able to locate only 4 of the members. These results also show the significance of the secondary structure for searching homologue in addition to the primary structure. We repeat the same experiment for RNA families, UPSK, Antizyme, Parecho, Corona-FSE, Corona-pk3 and IFN-gamma. In all cases, PAL locates all members as the topmost hits (See Figure 7(b)). We agree that Blast is not the most appropriate tool for comparison as other tools such as RSEARCH, and our own tool FastR can search for structural homologs of RNA [11, 26]. However, these other tools cannot align psuedoknotted RNA and the search must be followed up with a correct alignment to determine homologs. Also, the complexity of these methods often force a use of Blast to determine initial candidates. In the next section, we show that our tool used in conjunction with RNA filters can efficiently search large genomes.

9 Searching Genomes for Pseudoknots

While PAL is accurate in fishing for structural homologs, it is computationally intensive, making genome scale searches intractable. However, there has been much recent research (including our own work) on *computational filters* for RNA, which quickly eliminate much of the database, while retaining the true homologs [27, 26]. We used PAL in conjunction with sequence based filters [28] to search genomes, for the 3 most interesting families.

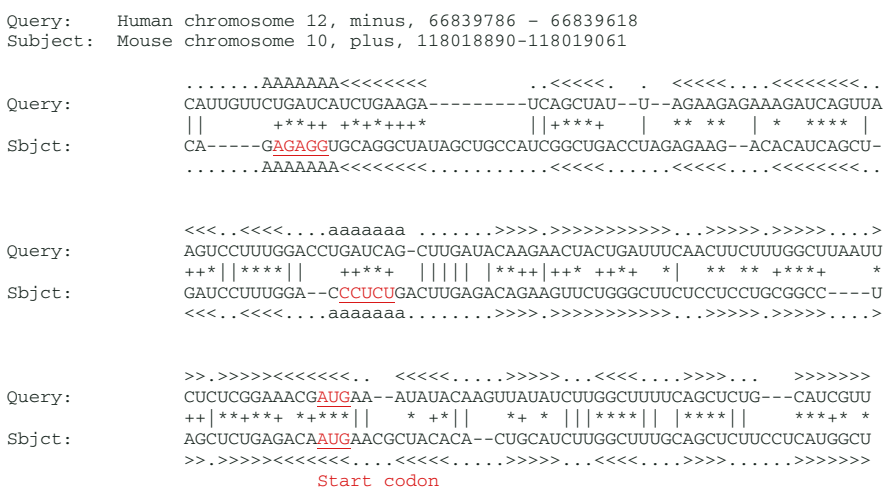


Fig. 8. Structural alignment of the Human Interferon-γ pseudoknot against mouse upstream genomic DNA. The structure of the query is denoted by parenthesis $<, >$" , and "A,a" for the pseudoknot. The symbols describe the conservation: (*) sequence and structure is conserved. (+) structure is conserved but not sequence. (|) sequence is conserved, but not structure.

The Corona-FSE family (RF00507) is a conserved pseudoknot in Coronaviruses which can promote ribosomal frameshifting [29]. We searched the entire Viral genome (79 Mb) for homologs of this family in 33.8 CPU hours on 1.6GHz AMD Opteron Grid, and identified 11 novel members of the sub-family. Like other known members, these are found in coronaviruses, murine hepatitis virus, and Avian flu viruses. Only 2 of the 11 were similar enough in sequence to be identified by BLAST. The alignments can be retrieved from (http://www.cse.ucsd.edu/∼bdost/RF00507.htm). A similar result was obtained for Corona-pk3. This family has a conserved $\sim$ 55nt pseudoknot structure which has been shown to be necessary for viral genome replication [30]. We identified 20 novel members of this family with significant scores (See http://www.cse.ucsd.edu/∼bdost/RF00165.htm). Only 1 of the 20 was similar enough in sequence to be identified by BLAST.

The Interferon-gamma family is an interesting example of a pseudoknot that is found in the 5'UTR of the Interferon-gamma gene. It regulates translation of the downstream gene by binding to the kinase PKR, a known regulator of IFN-gamma translation [31]. After its discovery in 2002, the pseudoknot was found to be conserved in many mammals. Its presence in rodents was speculated, but the homolog was not located. We searched in mouse and rat genomic DNA, and in the complete gene of gerbil. In all 3 species, we clearly identified the homologs as the top-scoring alignment. The alignment of human and mouse pseudoknots are shown in Figure 8. The conserved location in the two species, just upstream of the start codon, and conservation of key elements validates the hit. We are working with collaborators on experimental validation, and to locate more members of this family.

In conclusion, we demonstrate that the algorithm for aligning pseudoknots, implemented as PAL represents a viable tool for searching for novel homologs, and for struc-

tural inference. We hope that our tool will help increase the impact and influence of pseudoknotted RNA in cellular function. PAL and supplemental data are available upon request.

Acknowledgement

In this research, Zhang and Bafna are supported by National Science Foundation grant NSF-DBI:0516440.

References

1. Argaman, L., et al.: Novel small RNA-encoding genes in the intergenic regions of *Escherischia coli*. Curr. Biol. **11** (2001) 941–950
2. Novina, C.D., Sharp, P.A.: The RNAi revolution. Nature **430** (2004) 161–164 News.
3. Storz, G.: An expanding universe of noncoding RNAs. Science **296** (2002) 1260–1263
4. Vitreschak, A., Rodionov, D., Mironov, A., Gelfand, M.: Riboswitches: the oldest mechanism for the regulation of gene expression? Trends in Genetics **20** (2003) 44–50
5. Winkler, W.C., Breaker, R.R.: Genetic control by metabolite-binding riboswitches. Chembiochem **4** (2003) 1024–1032
6. Eddy, S.: Non-coding RNA genes and the modern RNA world. Nature Reviews in Genetics **2** (2001) 919–929
7. Jaeger, J., Turner, D., Zuker, M.: Improved prediction of secondary structures for RNA. Proceedings of the National Academy of Sciences **86** (1989) 7706–7710
8. Zuker, M., Sankoff, D.: RNA secondary structures and their prediction. Bull. Math. Biol. **46** (1984) 591–621
9. Rivas, E., Eddy, S.: Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. Bioinformatics **16** (2000) 583–605
10. Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S.R., Bateman, A.: Rfam: annotating non-coding RNAs in complete genomes. Nucleic Acids Res **33** (2005) 121–124
11. Klein, R., Eddy, S.: Rsearch: Finding homologs of single structured rna sequences. BMC Bioinformatics **4** (2003) 44
12. Rastogi, T., Beattie, T.L., Olive, J.E., Collins, R.A.: A long-range pseudoknot is required for activity of the Neurospora VS ribozyme. EMBO J **15** (1996) 2820–2825
13. 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** (2004) 45–50
14. Theimer, C.A., Blois, C.A., Feigon, J.: Structure of the human telomerase RNA pseudoknot reveals conserved tertiary interactions essential for function. Mol Cell **17** (2005) 671–682
15. Nixon, P.L., Rangan, A., Kim, Y.G., Rich, A., Hoffman, D.W., Hennig, M., Giedroc, D.P.: Solution structure of a luteoviral P1-P2 frameshifting mRNA pseudoknot. J Mol Biol **322** (2002) 621–633
16. Akutsu, T.: Dynamic programming algorithm for RNA secondary structure prediction with pseudoknots. Disc. Appl. Math. **104** (2000) 45–62
17. Dirks, R.M., Pierce, N.A.: A partition function algorithm for nucleic acid secondary structure including pseudoknots. J Comput Chem **24** (2003) 1664–1677
18. Evans, P.: Algorithms and Complexity for Annotated Sequence Analysis. PhD thesis, University of Victoria, Victoria BC, Canada (1964)
19. Jiang, T., Lin, G., Ma, B., Zhang, K.: A general edit distance between rna structures. Journal of Computational Biology **9** (2002) 371–388

20. Rivas, E., Eddy, S.: A Dynamic Programming Algorithm for RNA Structure Prediction Including Pseudoknots. Journal of Molecular Biology **285** (1999) 2053–2068
21. Condon, A., Davy, B., Rastegari, B., Tarrant, F., Zhao, S.: Classifying RNA Pseudoknotted Structures. Theoretical Computer Science **320** (2004) 35–50
22. Rastegari, B., Condon, A.: Linear time algorithm for parsing rna secondary structure. In: 5th Workshop on Algorithms in Bioinformatics (WABI). (2005)
23. Matsui, H., Sato, K., Sakakibara, Y.: Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures. Bioinformatics **21** (2005) 2611–2617
24. Bafna, V., Muthukrishnan, S., Ravi, R.: Computing similarity between RNA strings. Combinatorial Pattern Matching **937** (1995) 1–14
25. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: 10.3 Covariance models: SCFG-based RNA profiles. In: Biological Sequence Analysis. Cambridge University Press (1998)
26. Zhang, S., Hass, B., Eskin, E., Bafna, V.: Searching genomes for non-coding rna using fastr. IEEE Transactions on Computational Biology and Bioinformatics **2** (200) 366–379
27. Weinberg, Z., Ruzzo, W.L.: Faster genome annotation of non-coding rna families without loss of accuracy. In: Proceedings of the Annual Intl. Conference on Computational Biology (RECOMB). (2004)
28. Zhang, S., Borovok, I., Aharonowitz, Y., Sharan, R., Bafna, V.: A Sequence-Based Filtering Method for ncRNA Identification and its Application to Searching for Riboswitch Elements. Manuscript (2005)
29. Baranov, P.V., Henderson, C.M., Anderson, C.B., Gesteland, R.F., Atkins, J.F., Howard, M.T.: Programmed ribosomal frameshifting in decoding the SARS-CoV genome. Virology **332** (2005) 498–510
30. Williams, G.D., Chang, R.Y., Brian, D.A.: A phylogenetically conserved hairpin-type 3' untranslated region pseudoknot functions in coronavirus RNA replication. J Virol **73** (1999) 8349–8355
31. Ben-Asouli, Y., Banai, Y., Pel-Or, Y., Shir, A., Kaempfer, R.: Human interferon-gamma mRNA autoregulates its translation through a pseudoknot that activates the interferon-inducible protein kinase PKR. Cell **108** (2002) 221–232