MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **61** (2009) 789-816

ISSN 0340 - 6253

RNAComp: A new method for RNA secondary structure alignment

F. Zare-Mirakabad[a], M. Sadeghi[c,d,*] H. Ahrabian[a,b] and A. Nowzari-Dalini[a,b]

[a]Department of Bioinformatics, Institute of Biochemistry and Biophysics, University of Tehran, Tehran, Iran.

[b]Center of excellence in Biomathematics, School of Mathematics, Statistics, and Computer Science, Collage of Science, University of Tehran, Tehran, Iran.

[c]National Institute of Genetic Engineering and Biotechnology, Tehran, Iran.

[d]School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran.

(Received July 15, 2008)

Abstract

In this paper, a new algorithm for alignment of two RNA secondary structures without pseudoknots is presented. The algorithm is based on finding the longest common sub-structures between two RNA structures and special effort is devoted for aligning the beginning and the end parts of the existing stems (base pairs) in the secondary structures of two RNAs. The results of structure alignment of different types of RNA by this algorithm are obtained, and the result of this algorithm show more consistency with the models in the evolution rather than the other existing structure alignment algorithms.

1 Introduction

RNA is an important molecule and the investigation of RNA secondary structures is a challenging task in the molecular biology. It has been realized that RNA performs a wide range of functions in biological system. The mRNAs carry genetic information from DNA to ribosome, where protein is synthesized. Evolutionary conserved tRNAs [25]

* Corresponding author. Email: sadeghi@nrcgeb.ac.ir.

and rRNAs [6] carry out protein synthesis. Small nuclear RNAs [26] are important for the splicing of pre-mRNAs, and small nucleolar RNAs [12] act as guide RNAs for the modification of other RNA molecules. Small interfering RNA (siRNA) and microRNA (miRNA) play an important role in post-transcriptional gene silencing process [1]. The untranslated terminal regions (5′ or 3′-UTR) of mRNAs can contain regulatory motifs, such as cis-acting elements, which play a role for post-transcriptional gene regulation [22]. In fact, the function of many trans-acting non-coding RNAs and cis-acting RNA regulatory elements depend on the presence of motifs that are conserved both in structure and in sequence [21]. These facts have in turn highlighted the need for suitable algorithms and tools for the analysis and the comparison of RNA sequences and structures for motif discovery. In this manner, the first step in motif discovery of RNAs is comparison of RNA sequences and structures.

Currently, the most reliable method of inferring RNA secondary structure, analysis and the comparison of them is based on alignment algorithms that can be used to find common sequences and structures of RNAs [13, 14, 23]. These methods do not directly use base-paired and unpaired nucleotides. Instead loops and stems are used as the basic unit making it difficult to define the semantic meaning in the process of converting one RNA into another.

There are several ways to represent RNA structures and formulate corresponding similarity measures. One of them is to represent RNA secondary structures as (labeled or unlabeled) trees. There are several algorithms for computing the distance between two trees. Shapiro et al. [24] proposed to compare RNA secondary structures by using tree models. Constructing tree models is based on the idea that stems or helices dominantly stabilize the secondary structures, and comparing can be done by edit or alignment distance. The first efficient edition algorithm for the ordered rooted trees is due to Zhang and Shasha [28]. Later, Jaing et al. [11] introduced the tree alignment algorithm which is based on a common super-tree, and faster algorithm for similar tree is provided in [9]. Another model with more expressive edit operations on RNA structures, arc-annotated sequences, is introduced in [5] and further studied in [10, 19]. Wang and Zhang studied the similar consensus problem for trees [27]. Ma et al. [19] presented an algorithm for computing the similarity between two RNA molecule structures taking into account the

primary and secondary structures. Höchsmann et al. [7] used a clever tree representation of RNA and presented an algorithm for evaluation the local similarity in RNA secondary structures. After that, Hocfackor et al. [8] compared RNA secondary structures by aligning the corresponding base pairing probability matrices that use computed McCaskill's partition function algorithm [20].

There are other works, in which tree models were constructed to analyze the similarity of RNA secondary structures [10, 13, 14, 23]. Recently Liao et al. [16] proposed to use graph to represent RNA secondary structures and then derive some invariants from graph to compare RNA secondary structures. In [17], each secondary structure is transformed into a linear sequence and the standard and famous Lempel-Ziv algorithm [15] is employed for the similarity analysis. Popular tools for comparing RNA secondary structures with optimal alignment include RNAPdist [2], RNAdistance [24], and RNAforester [7].

In this paper we propose a novel algorithm, RNAComp, for the similarity analysis of RNA secondary structures without pseudoknots. In our approach, each secondary structure is transformed into a linear sequence. The linear sequence contains the information on the corresponding RNA secondary structures stems, loops (hairpin, bulge, interior, external), and single strands. With regard to the predefined criteria, two RNAs are scanned, and the common similar sub-structures between them are obtained. Also in the alignment algorithm the content of the base pairs (primary sequence) are also considered. Later the other non similar parts of two RNAs are aligned. Special effort is devoted in the algorithm for aligning the beginning and the end parts of the existing stems in the secondary structures of two RNAs. The results of structure alignment of different types of RNA obtained by this algorithm are compared with the well known tool RNAforester [7], and the results of this algorithm show more consistency with the real models in the evolution.

The rest of this paper is organized as follows. The method is given in Section 2 which is composed of basic definitions and notations followed by the structure alignment algorithm. In Section 3, the results are given. Finally, the discussion is presented in Section 4.

2 Methods

2.1 Basic Definitions and Notations

RNA is a molecule that is composed of 4 types of (ribo)nucleotides. Each nucleotide contains a phosphate group, a sugar group (ribose) and a nitrogenous base. The RNA *primary sequence* is formed by the linkage of the phosphate groups. The non-planar 5 member ribose ring connects the phosphate to the base. Each nucleotide in an RNA sequence is known by one of four different bases, adenine (A), cytosine (C), guanine (G) and uracil (U).

An RNA sequence is represented as $R = R[1]R[2]\dots R[n]$ where $|R| = n$ and $R[i] \in \{A, C, G, U\}$ for $1 \leq i \leq n$. When talking of an RNA secondary structure, we mean the RNA structure formed by stems, hairpin, bulge, interior, and external loops, and single strands, where stems represent double strand helices, hairpins represent nucleotide sequences connected to stems, and loops (bulges, interiors, and externals) show single structured regions. Therefore, a secondary structure contains a collection of stems, loops and single strands. These structures are shown in Figure 1. Formally, a stem is a set of consecutive base pairs such as $(R[i], R[j])$ for some $1 \leq i < j \leq n$, and also all two base pairs $(R[i], R[j])$ and $(R[i'], R[j'])$ such that $1 \leq i, i', j, j' \leq n$ have to satisfy these conditions: $i < i' < j' < j$ or $i < j < i' < j'$. A loop is a set of consecutive bases that is not in any stem.

The secondary structure of RNA $R = R[1]R[2]\dots R[n]$ can be defined as a linear sequence $S = S[1]S[2]\dots S[n]$ where for $1 \leq i \leq n$, $S[i] \in \{(,), H, B, I, E, F\}$. For any

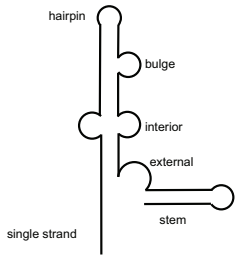

Figure 1: RNA Secondary structure contains stem, loop (hairpin, bulge, interior and external) and single strand.

base pair $(R[i], R[j])$ in a stem, $S[i] ='('$ and $S[j] =')'$ where $S[i]$ and $S[j]$ are the *begin-stem-element* and the *end-stem-element*. Loops are considered as four different types: hairpin, bulge, interior, and external. Therefore with regard to the structure of the RNA, $S[i] =' H'$ for the hairpin, $S[i] =' B'$ for the bulge, $S[i] =' I'$ for the interior, $S[i] =' E'$ for the external, and finally, $S[i] =' F'$ for the single strand nucleotides.

For example for the two RNAs given in Figure 2, with nucleotide sequence $R_1 = AACCUUGACCUUGUUCGGCUU$ and $R_2 = CAAAACCGAUCCGAUCUGGUUU$, the corresponding secondary structure sequences are $S_1 = ((((BB(((HHH))B)))B))$ and $S_2 = F((((BB(((HHH))))BB)))$, respectively.

For comparing the structure of two RNA sequences, we need a scoring scheme for similarity measurement. Here the scoring scheme is considered as a four parameters function $\delta(S_1[i], R_1[i], S_2[j], R_2[j])$, where $S_1[i]$ and $S_2[j]$ are structures of ith nucleotide and jth nucleotide of R_1 and R_2 respectively, and the definition of this function, for all $X, Y \in L = \{H, B, I, E, F\}$ is as follows.

$\delta(S_1[i], R_1[i], S_2[j], R_2[j]) =$

$$\begin{cases} SM_X & \text{if } R_1[i] = R_2[j],\ S_1[i] = S_2[j],\ S_1[i] = X, \\ SMIS_X & \text{if } R_1[i] \neq R_2[j],\ S_1[i] = S_2[j],\ S_1[i] = X, \\ SM_{SS} & \text{if } R_1[i] = R_2[j],\ S_1[i] = S_2[j],\ S_1[i], S_2[j] \in \{(,)\}, \\ SMIS_{SS} & \text{if } R_1[i] \neq R_2[j],\ S_1[i] = S_2[j],\ S_1[i], S_2[j] \in \{(,)\}, \\ SM_{XY} & \text{if } R_1[i] = R_2[j],\ S_1[i] \neq S_2[j],\ S_1[i] = X,\ \ S_2[j] = Y, \\ SMIS_{XY} & \text{if } R_1[i] \neq R_2[j],\ S_1[i] \neq S_2[j],\ S_1[i] = X,\ \ S_2[j] = Y, \\ SM_{SL} & \text{if } R_1[i] = R_2[j],\ S_1[i] \in L \text{ and } S_2[j] \in \{(,)\} \text{ or visversa}, \\ SMIS_{SL} & \text{if } R_1[i] \neq R_2[j],\ S_1[i] \in L \text{ and } S_2[j] \in \{(,)\} \text{ or visversa}, \\ SMNS & \text{if } S_1[i] \neq S_2[j],\ S_1[i], S_2[j] \in \{(,)\}, \end{cases}$$

where the values of $SM_X, SMIS_X, SM_{SS}, SMIS_{SS}, SM_{XY}, SMIS_{XY}, SM_{SL}, SMIS_{SL}$, and $SMNS$ should be set initially. The score of a nucleotide opposite to a gap is defined

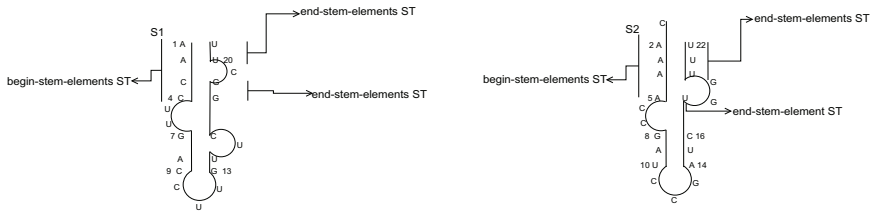

Figure 2: Two sample RNA structures R_1 and R_2.

as $\delta(S_1[i], R_1[i], -, -) = \delta(-, -, S_1[j], R_1[j]) = g$, where g is the score of a gap.

2.2 RNA alignment algorithm

In this section, preliminary a short explanation of the algorithm RNAComp is given, later the details are reviewed by an example. In this algorithm, first the similar sub-structures between two RNAs are obtained by a Dot-matrix. Later the selected similar sub-structures between two RNAs are scored with regard to their primary structure (type of nucleotides). The similar sub-structures with the highest score are selected and inserted into an alignment matrix. All the distances between similar sub-structures in the alignment matrix are filled with the simple alignment algorithm.

The steps of the algorithm RNAComp are summarized as follows.

1. Construct the Dot-Matrix D of size $n \times m$ by S_1 and S_2, where $|S_1| = n$ and $|S_2| = m$ and $n \leq m$. For $1 \leq i \leq n$ and $1 \leq j \leq m$ we have

$$D[i, j] \quad = \begin{cases} 1 & \text{if} \quad S_1[i] = S_2[j], \\ 0 & \text{otherwise.} \end{cases}$$

2. Similar sub-structures of length longer than a *cutoff* size are collected from matrix D and saved in the array *sub-similar-above-cutoff* and the remaining are saved in the array *sub-similar-below-cutoff* (the *cutoff* size is predefined due to the size of RNA), and all the sub-structures are scored with regard to the scoring schema δ.

3. Construct the alignment matrix M as follows:

   (a) Sort the elements of the array *Sub-similar-above-cutoff* with regard to their score.

   (b) Analyze the elements of *sub-similar-above-cutoff* into six different structures stem, hairpin, bulge, interior, external, and single strand.

   (c) Insert the analyzed structure elements with the length above the *cutoff* to the alignment matrix M if possible.

   (d) The analyzed structures with length below the *cutoff* are added to the array *sub-similar-below-cutoff*.

		C	A	A	A	A	C	C	G	A	U	C	C	G	A	U	C	U	G	G	U	U	U
		F	(	(	(	(	B	B	(	(	(	H	H	H	)	)	)	)	B	B	)	)	)
A	(	0	**1**	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
A	(	0	1	**1**	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
C	(	0	1	1	**1**	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
C	(	0	1	1	1	**1**	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
U	B	0	0	0	0	0	**1**	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
U	B	0	0	0	0	0	1	**1**	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
G	(	0	1	1	1	1	0	0	**1**	1	1	0	0	0	0	0	0	0	0	0	0	0	0
A	(	0	1	1	1	1	0	0	1	**1**	1	0	0	0	0	0	0	0	0	0	0	0	0
C	(	0	1	1	1	1	0	0	1	1	**1**	0	0	0	0	0	0	0	0	0	0	0	0
C	H	0	0	0	0	0	0	0	0	0	0	**1**	1	1	0	0	0	0	0	0	0	0	0
U	H	0	0	0	0	0	0	0	0	0	0	1	**1**	1	0	0	0	0	0	0	0	0	0
U	H	0	0	0	0	0	0	0	0	0	0	1	1	**1**	0	0	0	0	0	0	0	0	0
G	)	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	1	1	1	0	0	1	1	1
U	)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**1**	1	1	0	0	1	1	1
U	B	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
C	)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1
G	)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1
G	)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1
C	B	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
U	)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1
U	)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1

Figure 3: Dot-Matrix D.

(e) Sort the elements of the array *Sub-similar-below-cutoff* with regard to their score.

(f) Analyze the elements of *sub-similar-below-cutoff* into six different structures stem, hairpin, bulge, interior, external, and single strand.

(g) Insert the stem parts of elements to the alignment matrix M if possible.

4. In the alignment matrix M, all the alignments corresponding to the loops (hairpin, bulge, interior, external) and single strands are omitted.

5. The unaligned sections in the alignment matrix M are aligned by simple alignment algorithm.

6. Eventually, a simple path in the alignment matrix M from position $(0, 0)$ to (n, m) shows the alignment of the two RNAs.

Now the detail of the algorithm is illustrated with an example by two RNAs S_1 and S_2 given in Figure 2.

In Step 1, the Dot-Matrix D for two given RNAs in Figure 2 is constructed. This Matrix is shown in Figure 3. All the right-skewed sub-diagonals with consecutive values of 1 in the Dot-Matrix show the similar sub-structures. Obviously the matrix D has

maximum $m \times n$ sub-diagonals and each sub-diagonal shows consecutive number of 1's. One of these sub-diagonals is shown in bold in Figure 3. This sub-diagonal is called 1-diagonal and defines the similar sub-structures of two RNAs.

Then in Step 2, with regard to the content of the nucleotides, these similar sub-structures are scored. For any 1-diagonal from the positions p_1 and p_2 with size q in R_1 and R_2, we have

$$D[i,j] = 1 \text{ for } p_1 \leq i \leq p_1 + q - 1, \text{ and } p_2 \leq j \leq p_2 + q - 1,$$

and the score of this 1-diagonal is obtained by the following formula:

$$score = \sum_{k=0}^{q-1} \delta(S_1[p_1 + k], R_1[p_1 + k], S_2[p_2 + k], R_2[p_2 + k]).$$

Each 1-diagonal is denoted by $(p_1, p_2, q, score)$ and saved in the array *sub-similar-above-cutoff* if $q \geq cutoff$, otherwise saved in the array *sub-similar-below-cutoff*. For the given two RNAs in Figure 2 the elements of two arrays *sub-similar-above-cutoff* and *sub-similar-below-cutoff* for *cutoff*= 2 are shown in Figure 4 and Figure 5, respectively.

In Step 3, similar to the Dynamic Programming techniques, two RNA structures S_1 and S_2 are aligned and based on the produced similar sub-structures in Step 2, an alignment matrix M is constructed (S_1 as row and S_2 as column). Initially, the alignment matrix M is empty. The similar sub-structures (1-diagonals) of two RNA structures S_1 and S_2 from *sub-similar-above-cutoff* are considered one by one and inserted in the alignment matrix M with regard to the following convention.

First, all the 1-diagonals in the array *sub-similar-above-cutoff* are sorted with regard to their score. For inserting each 1-diagonal from this array to the alignment matrix, each 1-diagonal is partitioned to six different possible sub-structures: stem, hairpin, bulge,

no	$p_1, p_2, q, score$	no	$p_1, p_2, q, score$	no	$p_1, p_2, q, score$	no	$p_1, p_2, q, score$
1	1,2,14,204	11	2,2,3,26	21	13,21,2,18	31	13,15,2,16
2	5,18,2,96	12	16,14,3,24	22	1,9,2,18	32	7,9,2,16
3	16,15,4,72	13	2,8,3,24	23	13,20,2,18	33	11,11,2,16
4	15,19,4,72	14	1,4,2,20	24	10,12,2,18	34	8,8,2,16
5	19,19,3,68	15	20,21,2,20	25	16,16,2,18		
6	13,16,3,66	16	7,4,2,18	26	3,2,2,16		
7	1,3,3,28	17	20,16,2,18	27	17,14,2,16		
8	7,3,3,26	18	20,15,2,18	28	16,21,2,16		
9	7,2,3,26	19	20,14,2,18	29	3,8,2,16		
10	1,8,3,26	20	8,2,2,18	30	17,20,2,16		

Figure 4: Array *sub-similar-above-cutoff*

no	$p_1, p_2, q, score$	no	$p_1, p_2, q, score$	no	$p_1, p_2, q, score$
1	19,7,1,50	12	1,5,1,10	23	1,10,1,8
2	19,6,1,50	13	20,22,1,10	24	16,22,1,8
3	15,7,1,48	14	12,11,1,8	25	13,17,1,8
4	5,19,1,48	15	9,8,1,8	26	4,8,1,8
5	15,6,1,48	16	7,5,1,8	27	10,13,1,8
6	6,18,1,48	17	4,2,1,8	28	7,10,1,8
7	5,7,1,48	18	18,14,1,8	29	18,20,1,8
8	6,6,1,48	19	21,14,1,8	30	16,17,1,8
9	21,20,1,10	20	21,14,1,8	31	14,14,1,8
10	20,17,1,10	21	9,2,1,8		
11	14,20,1,10	22	13,22,1,8		

Figure 5: Array *sub-similar-below-cutoff*

		C	A	A	A	A	C	C	G	A	U	C	C	G	A	U	C	U	G	G	U	U	U
		F	(	(	(	(	B	B	(	(	(	H	H	H	)	)	)	)	B	B	)	)	)
A	(		s																				
A	(			s																			
C	(																						
C	(																						
U	B						l												*				
U	B							l												*			
G	(																						
A	(									s													
C	(										s												
C	H											l											
U	H												l										
U	H													l									
G	)														s								
U	)															s							
U	B																						
C	)																						
G	)																						
G	)																						
C	B																						
U	)																					s	
U	)																						s

Figure 6: Alignment Matrix M.

interior, external, and single strand. Then the possibility of inserting these sub-structures to the alignment matrix M are investigated.

For inserting a stem into the matrix M, the possibility of adding the *being-stem-elements* and *end-stem-elements* should be investigated simultaneously. In other word if the *begin-stem-elements* of a stem is aligned, we should search for their *end-stem-elements* and they are also aligned and both of them inserted into the alignment matrix M as shown in Figure 6.

As denoted in Figure 4, the longest similar sub-structure of length 14 between two RNAs R_1 and R_2 are located in positions 1 and 2, and is scored 204. The structure of this 1-diagonal is shown in Figure 7 separately. This 1-diagonal is analyzed due to its possible different structures: stem, hairpin, bulge, interior, external, and single strand.

	ST													
position in R_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	(	(	(	(	B	B	(	(	(	H	H	H	)	)
position in R_2	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 7: The longest 1-diagonal between R_1 and R_2.

During this separation, for each stem, the *begin-stem-elements* and *end-stem-elements* are considered simultaneously. Consecutive *begin-stem-elements* and *end-stem-elements* are defined by $(s, i_b, j_b, i_e, j_e, size)$ where s stands for stem, i_b and j_b denote the first position of *begin-stem-element* in this similar stem of R_1 and R_2 respectively and also i_e and j_e denote the first position of *end-stem-element* in this similar sub-structure of R_1 and R_2. Also *size* shows the number of base pairs in this stem. Respectively loops and single strands are defined by $(\ell, i_b, j_b, size)$ where ℓ stands for loop or single strand, i_b and j_b denotes the start position of loops (hairpin, bulge, interior, external) or single strands in R_1 and R_2 respectively, and *size* shows the number of nucleotides in the loop or in the single strand. The result of the analysis of the 1-diagonal $(1, 2, 14, 204)$ is given in Figure 8.

Note that the stem ST shown in Figure 7, is partitioned into three different stems, because the *end-stem-elements* of this stem in R_1 and R_2 are not consecutive (see Figure 2). Subsequently, the partitions are investigated for inserting in the alignment matrix M. All the partitions with a length above the *cutoff* are inserted to the alignment matrix. The analyzed stems with length less than *cutoff* are inserted to array *sub-similar-below-cutoff*. In this example, *sub-similar-below-cutoff* (shown in Figure 5) has 31 elements that currently the elements $(s, 3, 4, 18, 20, 1)$, $(s, 4, 5, 17, 17, 1)$ and $(s, 7, 8, 16, 16, 1)$ which are obtained from 1-diagonal $(1, 2, 14, 204)$ are inserted in the form $(3, 4, 1, 16)$, $(4, 5, 1, 16)$ and $(7, 8, 1, 20)$. It contains 34 elements.

Later, the other sorted 1-diagonals kept in the *sub-similar-above-cutoff* $[i]$ for $2 \leq i \leq$

$s, 1, 2, 20, 21, 2$
$s, 3, 4, 18, 20, 1$
$s, 4, 5, 17, 17, 1$
$\ell, 5, 6, 2$
$s, 7, 8, 16, 16, 1$
$s, 8, 9, 13, 14, 2$
$\ell, 10, 11, 3$

Figure 8: Result of analysis of the 1-diagonal $(1, 2, 14, 204)$.

|*sub-similar-above-cutoff*| are merged one by one to the alignment matrix. First, each element *sub-similar-above-cutoff* [*i*] is analyzed due to possible six different sub-structures (stem, hairpin, bulge, interior, external, and single strand), then the overlapping partitions of the inserted 1-diagonal with the 1-diagonal kept in the alignment matrix are omitted. For example the 1-diagonal $(5, 18, 2, 96)$ in Figure 4 is analyzed to one section. Since this section has the length above the *cutoff*, then it is inserted to the alignment matrix M. In Figure 6 this section are denoted by two $'*'$. As seen in this figure, the loop corresponding to this section in the entry $(5, 18)$ and $(6, 19)$ overlaps with the currently existing loop in the entry $(5, 6)$ and $(6, 7)$in the table. Therefore, this loop is not inserted.

Consistency in the order of the structures should be considered in inserting the 1-diagonal to the alignment matrix. As it is denoted in Figure 9, we consider a 1-diagonal in the alignment matrix that corresponds to the alignment of sub-structure P_1 in S_1 and sub-structure P_1' in S_2. By inserting the new 1-diagonal corresponding to the alignment P_2 in R_1 with P_2' in R_2 or P_2 in R_1 with P_3' in R_2, the consistency in orders of the structures is not violated. But on the other hand, if we consider the 1-diagonal in the alignment matrix corresponds to the alignment of sub-structure P_1 in R_1 and P_2' in R_2, then inserting the new 1-diagonal corresponding to the alignment P_2 in R_1 with P_1' in R_2 would violate the consistency, and we can just align P_2 in R_1 with P_3' in R_2.

After inserting all the possible sub-structures to the alignment matrix M, the next phase of the alignment starts. All the similar sub-structures in the *sub-similar-below-cutoff* are sorted in the decreasing order. In this step of the algorithm, similar to the above, all the similar sub-structures in *sub-similar-below-cutoff* are analyzed and inserted into the alignment matrix M. The only difference is that, in the analyzed similar sub-structures

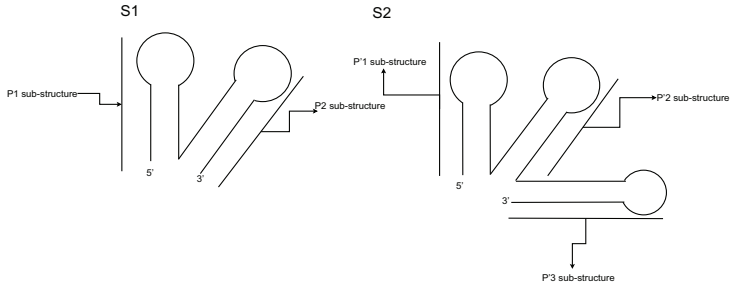

Figure 9: The structures S_1 and S_2.

of *sub-similar-below-cutoff*, only the stem sections are inserted into the alignment matrix (if possible).

In Step 4, all the assigned alignments on loops in the alignment matrix M are omitted. The reason for this process is that two loops with different sizes, for example a small loop with a bigger loop, can not be aligned properly with the structure alignment. But a simple alignment algorithm, can easily align those loops by inserting gaps in between.

In Step 5, all the empty regions between the sub-diagonals in matrix M, are aligned with a simple alignment algorithm and inserted to the matrix M. Assume there are two sub-diagonals from $(0,0)$ to (p_1,q_1) and from (p_2,q_2) to (p_3,q_3) in matrix M that show similar stems in these positions in two RNAs. The region in between, i.e. position (p_1+1,q_1+1) to (p_2-1,q_2-1) can be aligned as:

$$Align[i,j] \quad = Max \begin{cases} Align[i-1,j]+g, \\ Align[i,j-1]+g, \\ Align[i-1,j-1]+\delta(S_1[i],R_1[i],S_2[j],R_2[j]). \end{cases}$$

where $p_1+1 \leq i \leq p_2-1$ and $q_1+1 \leq j \leq q_2-1$. In this way, the empty distance between two consecutive diagonals in the matrix are filled.

Finally, we can obtain a path in the alignment matrix from position $(0,0)$ to the position (n,m). This path shows the alignment of two RNA structures.

3 Results and discussion

The source code of the algorithm described in this paper for computing optimal alignments of RNA secondary structures, has been written in perl and called RNAComp. We performed the algorithm with several RNaseP(Alcaligenes eutrophus, Streptomyces bikiniensis, Anacystis nidulans), tRNA (Alanine and Leucine of E.cloi), 5SRNA(Bacillus subtilis, Deinococcus Radiodurans), and Group I introns (Acanthamoeba griffini, Chlorella sorokiniana). These RNAs are taken from the RNaseP database [3], the Genomic tRNA database [18] and Gutell Lab Comparative RNA Web site [4]. The results of our algorithm in these sets are compared to the results of RNAforester algorithm [7].

As mentioned, the comparison of RNA structures is done by a scoring schema. Our scoring schema was given in Section 2. The constant values of this schema are demonstrated in Figure 10. The gap value is considered as -3 ($g = -3$) and *cutoff=1*. The RNAforester algorithm employs another scoring schema which is defined as follows,

$\sigma(B, B') = b_r$, $\sigma(B, -) = \sigma(-, B) = b_d$, $\sigma(P, P') = p_r$, $\sigma(P, -) = \sigma(-, P) = p_d$, and $\sigma(P, B) = \sigma(B, P) = \infty$, where B and B' stand for single base and P and P' means base-pair, and b_r, b_d, p_r, and p_d are constants [7]. The value of these constants are as follows: $p_r = 10$, $p_d = -5$, $b_d = -10$, and if $B = B'$ then $b_r = 1$, otherwise $b_r = 0$.

3.1 tRNA

The alignment results of the secondary structures of tRNA of Alanine and tRNA of Leucine of E.Coli obtained by RNAComp and RNAforester are presented in Figure 11 and Figure 12. As seen, there are scattered gaps in the alignment of RNAforester algorithm, but in RNAComp all the gaps are squeezed.

3.2 5SRNA

The alignment results of the 5SRNA secondary structure of Bacillus subtilis and Deinococcus Radiodurans obtained by RNAComp and RNAforester are presented in Figure 13 and Figure 14. Again scattered gaps are seen in the alignment of RNAforester algorithm result, but in RNAComp all the gaps are squeezed.

3.3 RNaseP

3.3.1 Alcaligenes eutrophus and Streptomyces bikiniensis

Figures 15 and 16 show the alignment results of the structures of RNaseP of Alcaligenes eutrophus (top) and Streptomyces bikiniensis (bottom) by RNAComp and RNAforester. As shown in these figures, some sub-structures of Streptomyces bikiniensis are deleted by RNAComp and RNAforester in comparison to Alcaligenes eutrophus. The deleted

$SM_H = 10$	$SM_I = 10$	$SM_F = 10$	$SM_E = 10$	$SM_B = 50$
$SMIS_H = 8$	$SMIS_I = 8$	$SMIS_F = 8$	$SMIS_E = 8$	$SMIS_B = 48$
$SM_{HI} = 2$	$SM_{HF} = 6$	$SM_{HE} = 6$	$SM_{HB} = 6$	$SM_{IF} = 6$
$SM_{IE} = 6$	$SM_{IB} = 6$	$SM_{FE} = 6$	$SM_{FB} = 6$	$SM_{EB} = 6$
$SMIS_{HI} = 1$	$SMIS_{HF} = 4$	$SMIS_{HE} = 4$	$SMIS_{HB} = 4$	$SMIS_{IF} = 4$
$SMIS_{IE} = 4$	$SMIS_{IB} = 4$	$SMIS_{FE} = 4$	$SMIS_{FB} = 4$	$SMIS_{EB} = 4$
$SM_{SS} = 10$	$SMIS_{SS} = 8$	$SM_{SL} = 6$	$SMIS_{SL} = 4$	$SMNS = -10000$

Figure 10: Scoring schema of RNAComp algorithm.

```
GGGGCUAUAGCUCAGCUGGGAG-AGCGCUUGCAUGGCAUGCAAGA---GG-----UCAGCGGUUCGAUCCCGCUUAGCUCCACCA
(((((((EE((((HHHHHHHH)-)))E(((((HHHHHHH)))))E---EE-----EE(((((HHHHHHH)))))))))))FFFF
(((((((EE(((HHHHHHHHHHH)))E(((((HHHHHHH))))E(((HHHH)))EE(((((HHHHHHH)))))))))))FFFF
GCCGAAGUGGCGAAAUCGGUAGACGCAGUUGAUUCAAAAUCAACCGUAGAAAUACGUGCCGGUUCGAGUCCGGCCUUCGGCACCA
```

Figure 11: Structure Alignment of Alanine tRNA and Leucine tRNA of E.Coli by RNAComp.

```
GGGGCUAUAGCUCAGCUGGGAG-AGCGCUUGCAUGGCAUGCAAGAG--G---U-C--AGCGGUUCGAUCCCGCUUAGCUCCACCA
(((((((EE((((HHHHHHHH)-)))E(((((HHHHHHH)))))EE--E---E-E--(((((HHHHHHH)))))))))))FFFF
(((((((EE(((HHHHHHHHHHH)))E(((((HHHHHHH))))E(((HHHH)))EE(((((HHHHHHH)))))))))))FFFF
GCCGAAGUGGCGAAAUCGGUAGACGCAGUUGAUUCAAAAUCAACCGUAGAAAUACGUGCCGGUUCGAGUCCGGCCUUCGGCACCA
```

Figure 12: Structure Alignment of Alanine tRNA and Leucine tRNA of E.Coli by RNAforester.

sub-structures are marked in these figures. As it is shown in these figures, RNAComp has completely deleted the marked sub-structures and RNAforester has partially deleted marked sub-structures and few nucleotides. These results are also seen in the secondary structure alignment of these two RNAs in Figures 17 and 18.

```
UG--CUUGGUGGCGAUAGCGAAGAGGUCACACCCGUUCCCAUACCGAACACGGAAGUUAAGCUCUUCAGCGCCGAUGGUAGUCGGG
FF--(((((((((EEEEE(((((((IIII(((((((HHHHHHHHHHHHH))))BB)))III))))))B))E(((((((II((((((
FF((((((((((EEEEE(((((((IIII(((((((HHHHHHHHHHHHH))))BB)))III))))))B))E(((((((II((((((
ACACCCCCGUGCCCAUAGCACUGUGGAACCACCCCACCCCAUGCCGAACUGGGUCGUGAAACACAGCAGCGCCAAUGAUACUCGGA

GGUUU-CCCCCUGUGAGAGUAGGACGCCGCCAAG--C-
((HHH-)))))))II)))))))EEE)))))))--F-
((HHHH)))))))II)))))))EEE)))))))))FF
CCGCAGGGUCCCGGAAAAGUCGGUCAGCGCGGGGGUUU
```

Figure 13: Structure Alignment of 5SRNA Bacillus subtilis and Deinococcus Radiodurans by RNAComp.

```
UGCUUGGUGG-CG-AUAGCGAAGAGGUCACACCCGUUCCCAUACCGAACACGGAAGUUAAGCUCUUCAGCGCCGAUGGUAGUCGGG
FF(((((((-EE-EEE(((((((IIII(((((((HHHHHHHHHHHHH))))BB)))III))))))B))E(((((((II((((((
FF((((((((((EEEEE(((((((IIII(((((((HHHHHHHHHHHHH))))BB)))III))))))B))E(((((((II((((((
ACACCCCCGUGCCCAUAGCACUGUGGAACCACCCCACCCCAUGCCGAACUGGGUCGUGAAACACAGCAGCGCCAAUGAUACUCGGA

GGUUU-CCCCCUGUGAGAGUAGGAC-G-CCGCCAAGC-
((HHH-)))))))II)))))))EE-E-)))))))F-
((HHHH)))))))II)))))))EEE)))))))))FF
CCGCAGGGUCCCGGAAAAGUCGGUCAGCGCGGGGGUUU
```

Figure 14: Structure Alignment of 5SRNA Bacillus subtilis and Deinococcus Radiodurans by RNAforester.

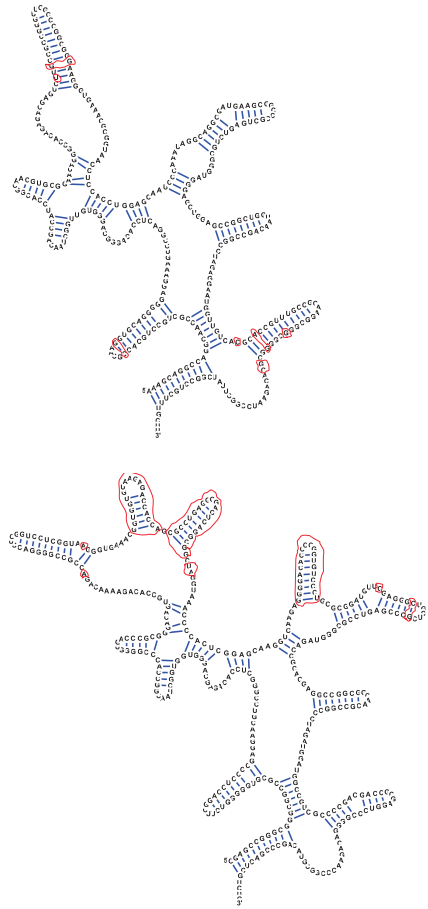

Figure 15: Aligned of Alcaligenes eutrophus and Streptomyces bikiniensis that are deleted by RNAComp.

3.3.2 Anacystis nidulans and Streptomyces bikiniensis

The alignment results of the secondary structures of RNAseP of Anacystis eutrophus and Streptomyces bikiniensis obtained by RNAComp and RNAforester are presented in Figure 19 and Figure 20. Again scattered gaps are seen in the alignment of RNAforester algorithm result, but in RNAComp all the gaps are squeezed.

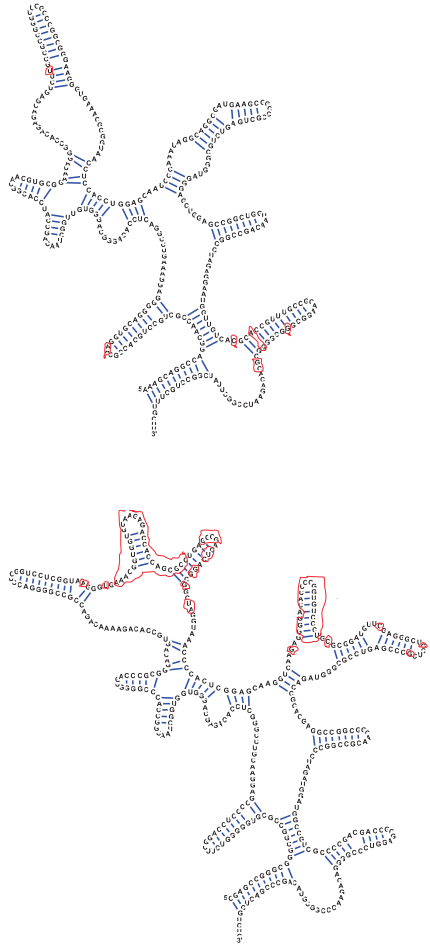

Figure 16: Aligned parts of Alcaligenes eutrophus and Streptomyces bikiniensis that are deleted by RNAforester.

3.3.3 Alcaligenes eutrophus and Anacystis nidulans

The alignment results of the secondary structure of RNAseP of Alcaligenes eutrophus and Anacystis nidulans obtained by RNAComp and RNAforester are presented in Figure 21 and Figure 22. Again scattered gaps are seen in the alignment of RNAforester algorithm result, but in RNAComp all the gaps are squeezed.

```
AAAGCAGGCCAGGCAACCGCUGCCUGCACCGCAAGGUGCAGGGGGAGGAAAGUCCGGACUCCACAGGGCAGGGUGUUGGCUAACAG
((((((((((((((((((E((I(((((((HHHH)))))))I))EEEEEEEEEEEEE((((BBBBBBBB(((((E((((HHHHH)
((((((((((((((((((E((I(((((((-HHHH-)))))))I))EEEEEEEEEEEEE((((BBBBBBBB(((((((((HHHHH)
CGAGCCGGGCGGGCGGCCGCGUGGGGGUC-UUCG-GACCUCCCCGAGGAACGUCCGGGCUCCACAGAGCAGGGUGGUGGCUAACGG

CCAUCCACGGCAACGUGCGGAAUAGGGCCACAGAGACGAG-UCUUGCCGCCGGGUUCGCCCGGCGGGAA-GGGUGAAAC-------
)))E(((((HHHH))))B)((BBB((IIIIIIIIIIIII-((((I(((((((HHHH)))))))I))-))IIIIIII-------
))))(((((HHHH))))B)((BBB((EEEEEEEEEEEEEEE((-(-(((((((HHHH)))))))-)BB))EEEEEEE(((((H
CCACCCGGGGUGACCCGCGGGACAGUGCCACAGAAAACAGACC-G-CCGGGGACCUCGGUCCUCGG-UAAGGGUGAAACGGUGGUG

-------------G---------------------C--GGUAACCUCCACCUGGAGCAAUCCCAAAU-------------------
-------------I---------------------I--IIIII)))))))))))))EEE((((IIII-------------------
HHHHHH))))))EE(((((((HHHH)))))))EEEEEEEE)))))))))))))EEE((((EEEE(((((((HHHH)))))))
UAAGAGACCACCAGCGCCUGAGGCGACUCAGGCGGCUAGGUAAACCCCACUCGGAGCAAGGUCAAGAGGGGACACCCCGGUGUCCC

-AGGCAGGCGAU-GAAGC-GGCCC-GCUGAGUCUGCGGGUAGGGAGCUGGAGCCGGCUGGUAACAGCCGGCCUAGAGGAAUGGUUG
-II((((((BBB-(((((-HHHHH-))))))))))IIIII))))EEEEEE(((((((HHHH)))))))EEEEEEEEEE)))))
)EE((((((BBBB((((((HHHHH))))))))))))EEEEE))))EEEEEE(((((((HHHH)))))))EEEEEEEEE)))))
UGCGCGGAUGUUCGAGGGCUGCUCGCCCGAGUCCGCGGGUAGACCGCACGAGGCCGGCGGCAACGCCGGCCCUAGAUGGAUGGCCG

UCACGCACCGUUUGCCGCAAGGCGGGCGGGGCGCACAGAAUCCGGCUUAUCGGCCUGCUUUGCUU
))EE((I(((((((((HHHH)))))))))I))EEEEEEEEEEEEEEEEE)B)))))))))FFFF
))E-((-(((II((((HHHH))))I-)))-))--EEEEEEEEEEEEEEE)B)))))))))FFFF
UCG-CC-CCGACGACCGCGAGGUCC-CGG-GG--ACAGAACCCGGCGUACAGCCCGACUCGUCUG
```

Figure 17: Structure Alignment of RNAseP Alcaligenes eutrophus and Streptomyces bikiniensis by RNAComp.

3.4 Group I Intron

The aligned parts of the secondary structures of Group I intron of Chlorealla sorokiniana and Acanthamoeba griffini obtained by RNAComp are drawn and presented in Figure 23. As shown in this figure, the sub-structures in Chlorealla sorokiniana labeled by A_1, B_1, C_1, $(D_1 + D_1')$, E_1, F_1, G_1, H_1, $(I_1 + I_1')$, J_1, K_1, L_1, M_1 and N_1 are aligned with sub-structures in Acanthamoeba griffini labeled by A, B, C, D, E, F, G, H, I, J, K, L, M and N respectively. Obviously in aligning two sub-structures of different lengths, gaps are employed. Also the unlabeled sub-structures in Acanthamoeba griffini and Chlorealla sorokiniana, are completely removed. Unfortunately, drawing the result of the alignment of RNAforester algorithm for these RNAs can not be done easily, because this algorithm keeps the number of gaps the least and this feature leads to the scattered gaps. As evidence we can see this point in the structure alignment of these two RNAs in Figure 24 and Figure 25.

```
AAAGCAGGCCAGGCAACCGCUGCCUGCACCGCAAGGUGCAGGGGGAGGAAAGUCCGGACUCCACAGGGCAGGGUGUUGGCUAACAG
((((((((((((((((((E((I(((((((HHHH)))))))I))EEEEEEEEEEEEE((((BBBBBBBB(((((E((((HHHHH)
((((((((((((((((((E((I(((((((HHH--H)))))))I))EEEEEEEEEEEEE((((BBBBBBBB(((((((((HHHHH)
CGAGCCGGGCGGGCGGCCGCGUGGGGGUCUUC--GGACCUCCCCCGAGGAACGUCCGGGCUCCACAGAGCAGGGUGGUGGCUAACGG

CCAUCCACGGCAACGUGCGGAAUAGGGCCACAGAGACGAGUCUUGCCGCCGGGUUCGCCCGGCGGGA-AGG-G-------------
)))E(((((HHHH))))B)((BBB((IIIIIIIIIIIII((((I(((((((HHHH))))))))I)-)))-I-------------
))))(((((HHHH))))B)((BBB((EEEEEEEEEEEEEEE((-(((((((((HHHH)))))))))BB))EEEEEEE(((((HHH
CCACCCGGGGUGACCCGCGGGACAGUGCCACAGAAAACAGACC-GCCGGGGACCUCGGUCCUCGGUAAGGGUGAAACGGUGGUGUA

----------------UGA----A---A--C-GC--GGUAACCUCCACCUGGAGCAAUCCCAA-AU---A---------------G
----------------III----I---I--I-II--IIIII)))))))))))))EEE((((II-II---I---------------I
HHHH)))))EE(((((((HHHH)))))))EEEEEEEEE)))))))))))))EEE((((EEEE(((((((HHHH)))))))E
AGAGACCACCAGCGCCUGAGGCGACUCAGGCGGCUAGGUAAACCCCACUCGGAGCAAGGUCAAGAGGGGACACCCCGGUGUCCCUG

-GCAGGCGAU-GAAGCG-GCCC-GCUGAGUCUGCGGGUAGGGAGCUGGAGCCGGCUGGUAACAGCCGGCCUAGAGGAAUGGUUGUC
-(((((BBB-(((((H-HHHH-))))))))))IIIII))))EEEEEE(((((((HHHH)))))))EEEEEEEEEE)))))))
E(((((BBBB((((((HHHHH))))))))))))EEEEE))))EEEEEE(((((((HHHH)))))))EEEEEEEEEE)))))))
CGCGGAUGUUCGAGGGCUGCUCGCCCGAGUCCGCGGGUAGACCGCACGAGGCCGGCGGCAACGCCGGCCCUAGAUGGAUGGCCGUC

ACGCACCGUUUGCCGCAAGGCGGGCGGGGCGCACAGAAUCCGGCUUAUCGGCCUGCUUUGCUU
EE((I(((((((((HHHH)))))))))I))EEEEEEEEEEEEEEEEE)B)))))))))FFFF
E-((-(((II((((HHHH))))-I)))-))--EEEEEEEEEEEEEEE)B)))))))))FFFF
G-CC-CCGACGACCGCGAGGUC-CCGG-GG--ACAGAACCCGGCGUACAGCCCGACUCGUCUG
```

Figure 18: Structure Alignment of RNAseP Alcaligenes eutrophus and Streptomyces bikiniensis by RNAforester.

4 Discussion

The RNA structures can be classified with regard to their consensus motifs. In fact consensus motifs are similar sub-structures which are common in two or more RNAs. For finding the consensus motifs, the comparison and alignment algorithms on the structure of the RNAs are needed. In this paper, we have presented a new algorithm, RNAComp, for alignment of two RNA secondary structures without pseudoknots. Our approach is based on finding common similar sub-structures between two RNAs with respect to the beginning and the end parts of the existing stems. Simultaneously nucleotide types in base pairs are considered. After aligning the similar sub-structures, the non-similar structures are also aligned with a simple sequence alignment algorithm.

In RNAComp algorithm, the two RNAs are compared in structural level, but in the some RNA comparison algorithms such as RNAforester [7], the comparison is done in nucleotide level. Alignment generated by these approach has many small gaps scattered throughout the structure. For this reason, the insertion and deletion (indel) operators in

```
GCGGGGAAAGGAGGCGAGGCAGUUGCGGCUCAGGCUUCGGUUAUGGGCUGAGGAAAGUCCGGGCUCCCAAAAGACCAGACUUGCUG
FF(((((((I(((B(((((I(((E((((((((((HHHH)))B)))))))EEEEEEEEEEEE(((((BBBBBBBBB(((B(((((
----((((((-(((((-(((((((E((I(((((((HHHH)))-))))I))EEEEEEEEEEEEE-((((BBB-BBBBB(((-(((((
----CGAGCC-GGGCG-GGCGGCCGCGUGGGGGUCUUCGGAC-CUCCCCGAGGAACGUCCGGG-CUCCACA-GAGCAGGG-UGGUG

GGUAACGCCCAGUGCGGGUGACCGUGAGGAGAGUGCCACAGAAA-CAUACCGCCGAUGGCCUGCUUGCAGGCACAGGUAAGGGUGC
((HHHHH)))))(((((HHHH))))B)((BBB((EEEEEEEEEE-EEEE(((((BB(((((((HHHH)))))B)))))BB))EEEE
((HHHHH)))))(((((HHHH))))B)((BBB((EEEEEEEEEEEEEEE(((((--((((((-HHHH-))))-)))))BB))EEEE
GCUAACGGCCACCCGGGGUGACCCGCGGGACAGUGCCACAGAAAACAGACCGCC--GGGGAC-CUCG-GUCC-UCGGUAAGGGUGA

AAGGGUGCGGUAAGAGCGCACCAGCAACAUC-GAGA-GGUGUUGGCUCGGUAAACCCCGGUUGGGAGCAAGGU-G-GA--------
EEE((((((HHHHHHH))))))EE(((((((-HHHH-)))))))EEEEEEEEE)))))))))))))EEE(((-I-II--------
EEE((((((HHHHHHH))))))EE(((((((((HHHH)))))))))EEEEEEEEE)))))))))))))-EEE((((EEEE(((((((
AACGGUGGUGUAAGAGACCACCAGCGCCUGAGGCGACUCAGGCGGCUAGGUAAACCCCACUCGGAG-CAAGGUCAAGAGGGGACAC

------------G--GGACAA--CGGUUGGUCUUUU--ACCUGUUCCGUUUAUGGACCGCUAGAGGUGGCUAGUAAUAGCCAUCCC
------------I--(((((B--B(((HHHHHHHHH--)))))))))IIIIIIIII)))EEEEEE(((((((HHHH)))))))EE
HHHH)))))))EE((((((BBBB((((((HHHHH))))))))))))EEEEE-)-)))EEEEEE(((((((HHHH)))))))EE
CCCGGUGUCCCUGCGCGGAUGUUCGAGGGCUGCUCGCCCGAGUCCGCGGGUA-G-ACCGCACGAGGCCGGCGGCAACGCCGGCCCU

AGAGAGAUAACAGCC-CUCUG--UC--UUC---GA-CAGAGAACAGAACCCGGCUUAUGU--CCUGCUUUCCCUACUUUAUUU
EEEEEEEE)))I)))-(((((--((--HHH---))-)))))EEEEEEEEEEEEEEEE)B)--))))I)))))))FFFFFFFFF
EEEEEEEE)))))))E(((((II((((HHHH))))I)))))E-EEEEEEEEEEEEEE--)B))))-))))))--FFFF-----
AGAUGGAUGGCCGUCGCCCCGACGACCGCGAGGUCCCGGGGA-CAGAACCCGGCGUA--CAGCCC-GACUCG--UCUG-----
```

Figure 19: Structure Alignment of RNAseP Anacystis eutrophus and Streptomyces bikiniensis by RNAComp.

our algorithm are based on insertion or deletion of sub-structures. This approach avoids small gaps and causes multiple residue indel events which lead to fewer longer gaps, and is similar to the results obtained by affine gap penalty in the sequence alignment. Based on the evolutionary model a single mutation event can insert or delete multiple residues and therefore the sub-structure insertion and deletion occur more than the scattered nucleotide insertion and deletion in a family of RNA sequences.

The RNAComp algorithm is performed and tested on different families of RNAs and the obtained results are compared with RNAforester algorithm. Alignment of RNA structures of different families shows that the obtained results by RNAComp are consistent with evolution rather than RNAforester algorithm.

```
GCGGGGAAAGGAGGCGAGGCAGUUGCGGCUCAGGCUUCGGUUAUGGGCUGAGGAAAGUCCGGGCUCCCAAAAGACCAGACUUGCUG
FF(((((((I(((B(((((I(((E((((((((((HHHH)))B)))))))EEEEEEEEEEEEE(((((BBBBBBBBB(((B(((((
--(((((((-((--(-(((((((E((I(((((((HHHH)))-))))I))EEEEEEEEEEEEE((((-BBB-BBBBB(((-(((((
--CGAGCCGG-GC--G-GGCGGCCGCGUGGGGGUCUUCGGAC-CUCCCCGAGGAACGUCCGGGCUCC-ACA-GAGCAGGG-UGGUG

GGUAACGCCCAGUGCGGGUGACCGUGAGGAGAGUGCCACAGAAA-CAUACCGCCGAUGGCCUGCUUGCAGGCACAGGUAAGGGUGC
((HHHHH)))))(((((HHHH))))B)((BBB((EEEEEEEEEE-EEEE(((((BB(((((((HHHH)))))B)))))BB))EEEE
((HHHHH)))))(((((HHHH))))B)((BBB((EEEEEEEEEEEEEEE(((((--((((((-HHHH-))))-)))))BB))EEEE
GCUAACGGCCACCCGGGGUGACCCGCGGGACAGUGCCACAGAAAACAGACCGCC--GGGGAC-CUCG-GUCC-UCGGUAAGGGUGA

AAGGGUGCGGUAAGAGCGCACCAGCAACAUCGA-GA-GGUGUUGGCUCGGUAAACCCCGGUUGGGAGCAAGGU---G-G---A---
EEE((((((HHHHHHH))))))EE(((((((HH-HH-)))))))EEEEEEEEE)))))))))))))EEE(((---I-I---I---
EEE((((((HHHHHHH))))))EE(((((((HHHH)))))))EEEEEEEEE)))))))))-))))EEE((((EEEE(((((((
AACGGUGGUGUAAGAGACCACCAGCGCCUGAGGCGACUCAGGCGGCUAGGUAAACCCCACUC-GGAGCAAGGUCAAGAGGGGACAC

------------G--GGACAA--CGGUUGGU-CUU-UUACCUGUUCCGUUUAUGGACCGCUAGAGGUGGCUAGUAAUAGCCAUCCC
------------I--((((B--B(((HHHH-HHH-HH)))))))IIIIIIIII)))EEEEEE(((((((HHHH)))))))EE
HHHH)))))))EE((((((BBBB((((((HHHHH))))))))))))EEEEE-)-)))EEEEEE(((((((HHHH)))))))EE
CCCGGUGUCCCUGCGCGGAUGUUCGAGGGCUGCUCGCCCGAGUCCGCGGGUA-G-ACCGCACGAGGCCGGCGGCAACGCCGGCCCU

AGAGAGAUAACAGCC---CUC--UGUCUUC-GACA-GAGAACAGAACCCGGCUUAUGUCCUGCUUUCCCUACUUUAUUU
EEEEEEEE)))I)))---(((--((((HHH-))))-)))EEEEEEEEEEEEEEEE)B))))I)))))))FFFFFFFFF
EEEEEEEE)))))))E(((((II((((HHHH))))I)))))EEEEEEEEEEEEE-EE)B))-)))))))FFFF-----
AGAUGGAUGGCCGUCGCCCCGACGACCGCGAGGUCCCGGGGACAGAACCCGGCG-UACAGC-CCGACUCGUCUG-----
```

Figure 20: Structure Alignment of RNAseP Anacystis eutrophus and Streptomyces bikiniensis by RNAforester.

```
----AAAGCA-GGC-CAGGCAACCGCUGCCUGCACCGCAAGGU-GCAGGGGGAGGAAAGUCCGGA-CUCCACAGGGC-AGGG-UGU
----((((((-(((-(((((((((E((I(((((((HHHH)))-)))))I))EEEEEEEEEEEEE-((((BBBBBBB-B(((-((E
FF(((((((I(((B(((((I(((E((-(((((((HHHH)))B)))))-))EEEEEEEEEEEEE(((((BBBBBBBBB(((B(((
GCGGGGAAAGGAGGCGAGGCAGUUGCG-GCUCAGGCUUCGGUUAUGGGC-UGAGGAAAGUCCGGGCUCCCAAAAGACCAGACUUGC

UGGCUAACAGCCAUCCACGGCAACGUGCGGAAUAGGGCCACAGAGACGAG---UCUUGCCGCCGGGUUCGCCCGG-CGGGAAGG--
((((HHHHH))))E(((((HHHH))))B)((BBB((IIIIIIIIIIIII---((((I(((((((HHHH)))))-)))I))))--
((((HHHHH)))))(((((HHHH))))B)((BBB((EEEEEEEEEEEEE((((--BB(((((((HHHH)))))B))----)))B
UGGGUAACGCCCAGUGCGGGUGACCGUGAGGAGAGUGCCACAGAAACAUACCGCC--GAUGGCCUGCUUGCAGGCACA----GGUA

---GUG-AA--------------------AC------------------GC--GGUAACCUCCACCUGGAG-CAAUCCCAAAUAGG
---III-II--------------------II------------------II--IIIII)))))))))))))-EEE((((IIIIII(
B))EEEEEEE((((((HHHHHHH))))))EE(((((((HHHH)))))))EEEEEEEEE))))))))))))))EEE(((-III--I-
AGGGUGCAAGGGUGCGGUAAGAGCGCACCAGCAACAUCGAGAGGUGUUGGCUCGGUAAACCCCGGUUGGGAGCAAGGU-GGA--G-

CAGGCGAUGAAGCGGCCCGCUGAGUCUGCGGGUA-G-GGAGCUGGAGCCGGCUGGUAACAGCCGGCCUAGAGGAAUGGUUGUCACG
(((((BBB(((((HHHHH)))))))))))IIIII-)-)))EEEEEE(((((((HHHH)))))))EEEEEEEEE)))))))EE(
(((((-BB(((HHHHHHHHH)))))))IIIIIIIII)))EEEEEE(((((((HHHH)))))))EEEEEEEEE)))I)))--(
GGACA-ACGGUUGGUCUUUUACCUGUUCCGUUUAUGGACCGCUAGAGGUGGCUAGUAAUAGCCAUCCCAGAGAGAUAACAGCC--C

CACCGUUUGCCGCAAGGCGGGCGGGGCGCACAGAAUCCGGCUUAUCGGCC-UGCUUU--GCUU-----
(I(((((((((HHHH)))))))))I))EEEEEEEEEEEEEEEEE)B))))-))))))--FFFF-----
(-(((((----HHH-----)))))-))E-EEEEEEEEEEEEEEE)B))))I)))))))FFFFFFFFF
U-CUGUC----UUC-----GACAG-AGA-ACAGAACCCGGCUUAUGUCCUGCUUUCCCUACUUUAUUU
```

Figure 21: Structure Alignment of RNAseP Alcaligenes eutrophus and Anacystis eutrophus by RNAComp.

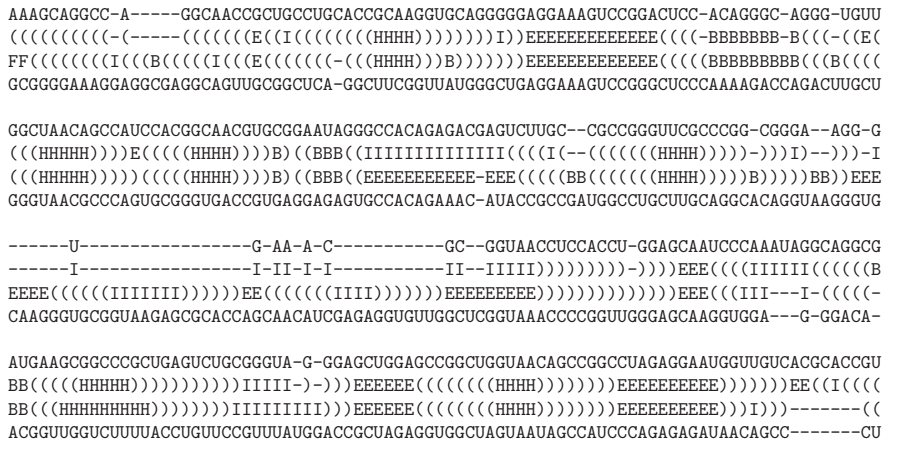

Figure 22: Structure Alignment of RNAseP Alcaligenes eutrophus and Anacystis eutrophus by RNAforester.

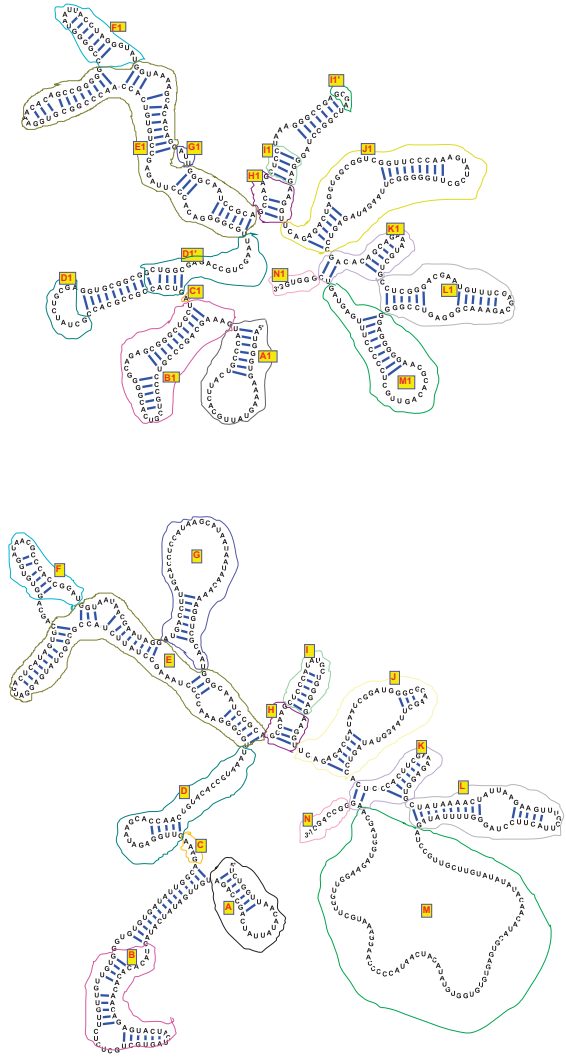

Figure 23: Aligned parts of Chlorealla sorokiniana and Acanthamoeba griffini by RNA-Comp.

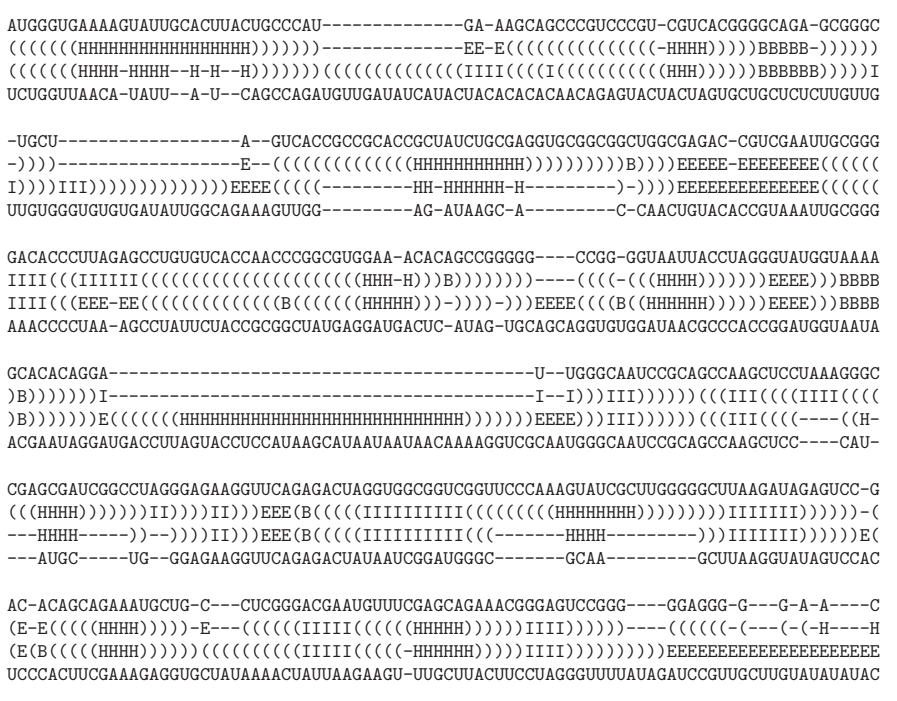

Figure 24: Structure Alignment of Group Intron I of Chlorealla sorokiniana and Acanthamoeba griffini by RNAComp.

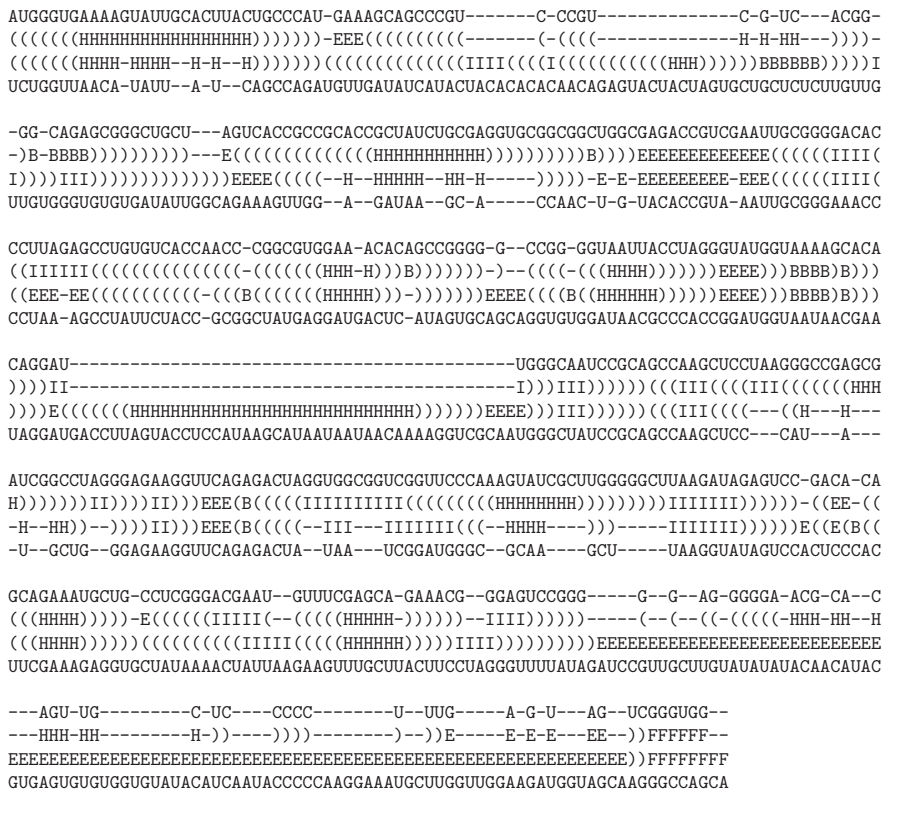

Figure 25: Structure Alignment of Group Intron I of Chlorealla sorokiniana and Acanthamoeba griffini by RNAforester

References

[1] V. Ambros, B. Bartel, D.P. Bartel, C.B. Berge, J.C. Carrington, X. Chen, G. Dreyfuss, S.R. Eddy, S. Griffiths-Jones, M. Marshall, M. Matzke, G. Euvkun, and T. Tuschl, A uniform system for microRNA annotation, *RNA* **9** (2003), 277–279.

[2] S. Bonhoeffer, J. S. McCaskill, P. F. Stadler, and P. Schuster, RNA multi-structure landscapes, *Euro. Biophys J.* **22** (1993), 13–24.

[3] J. Brown, The Ribonuclease P database, *Nucl. Acids Res.* **27** (1999), 314.

[4] J. Cannone, S. Subramanian, M. Schnare, J. Collett, L. DSouza, Y. Du, B. Feng, N. Lin, L. Madabusi, K. Muller, N. Pande, Z. Shang, N. Yu, and R. Gutell, The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs, *BMC Bioinformatics* **3** (2002), 15–45.

[5] P. Evans, *Algorithms and Complexity for Annotated Sequences Analysis*, Ph.D. thesis, University of Victoria, 1999.

[6] R. Gutell, N. Larsen, and C. Woese, Lessons from an evolving rRNA: 16s and 23s rRNA strucures from comparitive perspective, *Microbiol. Rev.* **58** (1994), 10–26.

[7] M. Höchsmann, T. Töller, R. Giegerich, and S. Kurtz, Local similarity in RNA secondary structures, in: *Proc. 2nd IEEE Computer Society Bioinformatics Conference* (Stanford, CA, USA), IEEE Computer Society Press, 11-14 August 2003, pp. 159–168.

[8] I. L. Hofacker, S. H. F. Bernhart, and P. F. Stadler, Alignment of RNA base pairing probability matrices, *Bioinformatics* **20** (2004), 2222–2227.

[9] J. Jansson and A. Lingas, A fast algorithm for optimal alignment between similar ordered trees, *Fund. Inform.* **56** (2003), 105–120.

[10] T. Jiang, G. H. Lin, B. Ma, and K. Zhang, A general edit distance between RNA structures, *J. Compu. Biology* **9** (2002), 371–388.

[11] T. Jiang, L. Wang, and K. Zhang, Alignment of trees - an alternative to tree edit, *Theoret. Comput. Sci.* **143** (1995), 137–148.

[12] T. Kiss, Small nucleolar RNAs: An abundant group of non coding RNAs with diverse cellular functions, *Cell* **109** (2002), 145–148.

[13] S.Y. Le, R. Nussinov, and J.V. Mazel, Tree graphs of RNA secondarystructures and their comparisons, *Comput. Biomed. Res.* **22** (1989), 461–473.

[14] S.Y. Le, J. Owens, R. Nussinov, J.H. Chen, B. Shapiro, and J.V. Mazel, RNA secondary structures: comparisons and determination of frequently recurring substructures by consensus, *Comput. Appl. Biosci.* **5** (1989), 205–210.

[15] A. Lempel and J. Ziv, On the complexity of finite sequences, *IEEE Trans. Inform. Theory* **22** (1976), 75–81.

[16] B. Liao and T. M. Wang, A 3D graphical representation of RNA secondary structure, *J. Biomol. Struc. Dynamics* **21** (2004), 827–832.

[17] N. Liu and T. Wang, A method for rapid similarity analysis of RNA secondary structures, *BMC Bioinformatics* **7** (2006), 1–11.

[18] T.M. Lowe and S.R. Eddy, tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence, *Nucl. Acids Res.* **25** (1997), 955964.

[19] B. Ma, L. Wang, and K. Zhang, Computing similarity between RNA structures, *Theoret. Comput. Sci.* **276** (2002), 111–132.

[20] J. S. McCaskill, The equilibrium partition function and base pair binding probabilities for RNA secondary structure, *Biopolymers* **29** (1990), 1105–1119.

[21] G. Pavesi, G. Mauri1, M. Stefani1, and G. Pesole, RNAProfile: An algorithm for finding conserved secondary structure motifs in unaligned RNA sequences, *Nucleic Acids Res.* **32** (2004), 3258–3269.

[22] G. Pesole, S. Liuni, G. Grillo, F. Licciulli, F. Mignone, C. Gissi, and C. Saccone, UTRdb abd UTRsite: specialized databases of sequences and functional elements of

5' and 3' untranslated regions of eukaryotic mRNAs, *Nucleic Acids Res.* **30** (2002), 335–340.

[23] B. Shapiro, An algorithm for comparing multiple RNA secondary structures, *Comput. Appl. Biosci.* **4** (1988), 387–393.

[24] B. Shapiro and K. Zhang, Comparing multiple RNA secondary structures using tree comparisons, *Comput. Appl. Biosci.* **6** (1990), 309–318.

[25] M. Sprinzl, C. Horn, M. Brown, A. Ioudovitch, and S. Steinberg, Compilation of tRNA sequences and sequences of tRNA genes, *Nucleic Acids Res.* **26** (1998), 148–153.

[26] T. Villa, J. Pleiss, and C. Guthrie, Spliceosomal snRNAs: mg(2+)- dependent chemistry at the catalytic core?, *Cell* **109** (2002), 149–152.

[27] J. Wang and K. Zhang, Identifying consensus of trees through alignment, *Inform. Sci.* **126** (2000), 165–189.

[28] K. Zhang and D. Shasha, Simple fast algorithms for the editing distance between trees and related problems, *SIAM J. Comput.* **18** (1989), 1245–1262.