Computing Similarity Between RNA Secondary Structures

Kaizhong Zhang*
Department of Computer Science
University of Western Ontario
London, Ont. N6A 5B7
Canada
kzhang@csd.uwo.ca

Abstract

The primary structure of a ribonucleic acid (RNA) molecule is a sequence of nucleotides (bases) over the four-letter alphabet $\{A, C, G, U\}$. The secondary structure of an RNA is a set of base-pairs (nucleotide pairs) which formed bonds between $A-U$ and $C-G$. These bonds have been traditional assumed to be non-crossing in the secondary structure. This implies a tree representation of the secondary structure of RNA molecule.

This paper considers several notions of similarity between two RNA molecule structures taking into account both the primary and the secondary structures. We consider a natural tree representation with both primary and secondary structure data. We present efficient algorithms for comparing such tree representation. We then show that some of these similarity notions can be used to solve the structure prediction problem when the structure of a closely related RNA is known.

keywords: Molecular biology, RNA secondary structures, Similarity and prediction.

1 Introduction

It is now a known fact that secondary and tertiary structural features of RNAs are important in the molecular mechanism involving their functions. The presumption, of course, is that to a preserved function there corresponds a preserved molecular confirmation and, therefore, a preserved secondary and tertiary structure. Therefore the ability to compare RNA secondary structures and to infer a secondary structure for a new RNA from a known RNA secondary structure is useful.

In RNA secondary structure, a bonded pair of bases (base-pair) is usually represented as an edge between the two complementary bases involved in the bond. It is assumed that any base participates in at most one such pair and the edges of the bonded pairs are non-crossing. This gives a tree-like structure as in Figure 1.

The similarity between two sequences have been formulated as problems of determining minimum cost edit distance, fining the optimal alignment, and computing a longest common subsequence (LCS) or calculating a shortest common super-sequence (SCS) [7, 10, 11]. We formulate the corresponding versions of these problems between two RNA structures. We also consider the problem of inferring secondary structure from the primary structure of an RNA when a closely related RNA's secondary structure is given.

Results

In table 1, we list the problems addressed in this paper and the time complexity of the algorithms we propose. In the table, n_1 and n_2 are the sizes of the primary structures of the two RNAs. dp_i, dg_i, and lp_i are factors which are much smaller than the corresponding size n_i.

The first two entries are dealing with edit or alignment distances which are the generalization of sequence edit and alignment distances. The third en-

*Research supported partially by the Natural Sciences and Engineering Research Council of Canada under Grant No. OGP0046373.

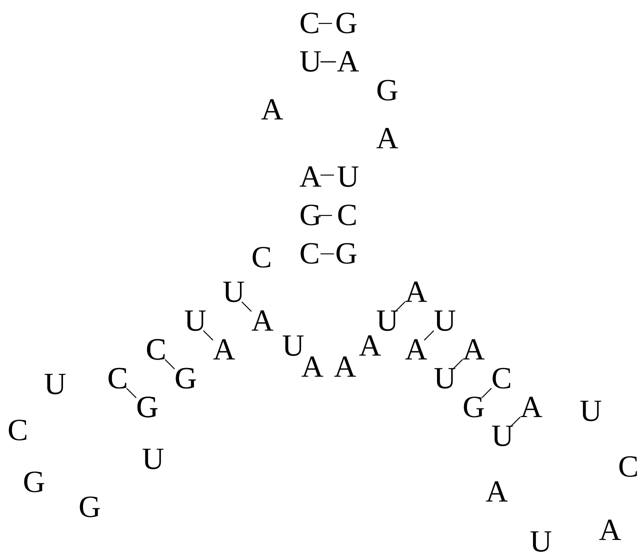

Figure 1: Secondary structure of an RNA.

	Problem	running time
Similarity	RNA edit distance	$O(n_1 n_2 dp_1 dp_2)$
	RNA alignment distance	$O(n_1 n_2 (dg_1 + dg_2)^2)$
	RNA Weighted Largest Common Sub-structure (WLCS)	$O(n_1 n_2 dp_1 dp_2)$
	RNA Weighted Smallest Common Super-structure (WSCS)	$O(n_1 n_2 (dg_1 + dg_2)^2)$
Prediction	inferring through WLCS	$O(n_1 n_2^2 + lp_1 n_2^3)$
	inferring through WSCS	$O(n_1 n_2^2 + lp_1 n_2^3)$

Table 1: Results

try is dealing with the problem of finding the largest common sub-structures of two RNA structures. The fourth problem is to find the smallest common super-structure, or equivalently the largest common sub-structure that comes from a super structure. Since applications of these algorithms usually require approximate solutions, we consider the weighted version of these problems.

Prediction of secondary structure of a single RNA from its primary structure has been widely studied [6, 13, 17]. The prediction considered here deal with the problem where two input RNAs are given such that the first one has primary and secondary structures and the other has only primary structure, and the goal is to find a secondary structure for the second RNA so that the WLCS (weighted largest common substructure) or WSCS measure is maximized. This method is useful to generate common folding structure from RNAs when there is extensive divergence of their primary structures.

Related work

Since the secondary structure appears as tree-like structure, there are works considering comparison using tree comparison [8, 4, 5, 9]. However these methods do not directly use base-paired nucleotides and unpaired nucleotides. Instead loops and stems (stacked pairs) are used as the basic unit making it difficult to define the semantic meaning in the process of converting one RNA into another. To overcome this difficulty, the method we propose in this paper directly use base-paired and unpaired nucleotides in the tree representation and apply some basic operations on them.

Another line of works are primary structure based where the comparison is basically done on the primary structure while trying to incorporate secondary structure data [1, 2]. The weakness of this approach is that it does not treat a base-pair as a whole entity. For example, in the comparison of two RNAs, a base-pair from one RNA can have one nucleotide deleted while the other nucleotide matched to nucleotide (unpaired or even paired) in the other RNA. Our method treat base-pair as a unit, it can be matched to another base-pair, it can be deleted, or it can be inserted. This is closer to the spirit of the comparative analysis method currently being used in the analysis of RNA secondary structures either manually or automatically.

2 Comparing two RNA secondary structures

2.1 RNA structure and basic operations

The primary structure of a ribonucleic acid (RNA) molecule is a sequence of nucleotides (bases) over the four-letter alphabet $\sum = \{A, C, G, U\}$. The secondary structure of an RNA is a set S of base-pairs (nucleotide pairs) which formed bonds between $A-U$ and $C-G$. When there is no confusion we will refer to secondary structure of an RNA as both its set S and its underline nucleotide sequence. Following Zuker [15, 16, 17], we assume a model where there is no knots in the secondary structure. This means that the bonds in S are non-crossing.

Following the tradition in sequence comparison [7, 10, 11], we define three operations, relabel, delete, and insert, on RNA secondary structure. For a given RNA secondary structure, each operation can be applied to either a base-pair in S or an unpaired base. Relabel a base-pair is to replace one element in S with another. This means that at the sequence level, two bases may be changed at the same time. Delete a base-pair is to delete the pair from S. At the sequence level, this means to delete two bases at the same time. Insert a base-pair is to insert a new element into S. At the sequence level, this means to insert two bases at the same time. This also implies that the after the insertion, elements in S have no conflict (non-crossing). Relabel an unpaired base is to replace it with another base. Delete an unpaired base is to delete the base from the sequence. Insert a base is to insert a new base into the sequence as an unpaired base.

We assume that there is a score function associate with the operations. The score function for base-pairs is defined on $\sum \times \sum \cup \{\lambda\}$, and the score function for unpaired bases is defined on $\sum \cup \{\lambda\}$.

With these definitions, we can consider how to translate one secondary structure into another using minimum number of weighted operations. This gives us a distance or (dis)similarity measure between two RNAs.

2.2 Tree representation

Recall that a secondary structure is denoted by the set S of all base-pairs that formed bonds. For $(i, j) \in S$, h is accessible from (i, j) if $i < h < j$ and there is no pair $(k, l) \in S$ such that $i < k < h < l < j$. Define (i, j) as the parent of $(k, l) \in S$ if k, l are accessible from (i, j). Define (i, j) as the parent of $h \notin S$ if h are accessible from (i, j). Note that each base-pair $(i, j) \in S$ and each unpaired base h has at most one parent, implying a tree (sometimes forest) on the elements of secondary structure. The definitions of child, sibling, and leaf follow naturally. The order imposed based upon the 5' to 3' nature of the molecule makes the tree an ordered tree, see Figure 2.

Following [12, 14], we can consider the tree edit operations. Relabeling a node means changing the label of the node. Deleting a node means making the children of node n become the children of the parent of n and then removing n. Inserting is the complement of deleting.

Examining each of the operations defined on RNA secondary structures, we can see that they are exactly the same as the tree edit operations defined on the tree representation. For example, consider the deletions of three base-pairs ((A, U), (G, C), and (C, G)) from the RNA in Figure 1 the tree representation of the resulting RNA is shown in Figure 3. Deleting the three corresponding nodes from Figure 2 using tree operation, we get the same tree.

Conversely most edit operations defined on the above tree representation are meaningful operations on RNA secondary structures. Theoretically there can be operations that dose not result in a valid secondary structure, i.e. inserting an unpaired base as an internal node, but we can show that the minimum cost sequence of tree edit operations that translate one tree into another will not use any of this kind operations.

Therefore we can use tree edit algorithms on this tree representation to compare two RNA secondary structures.

2.3 Algorithms for edit distance and alignment distance

We will consider several notions of similarity between RNA secondary structures.

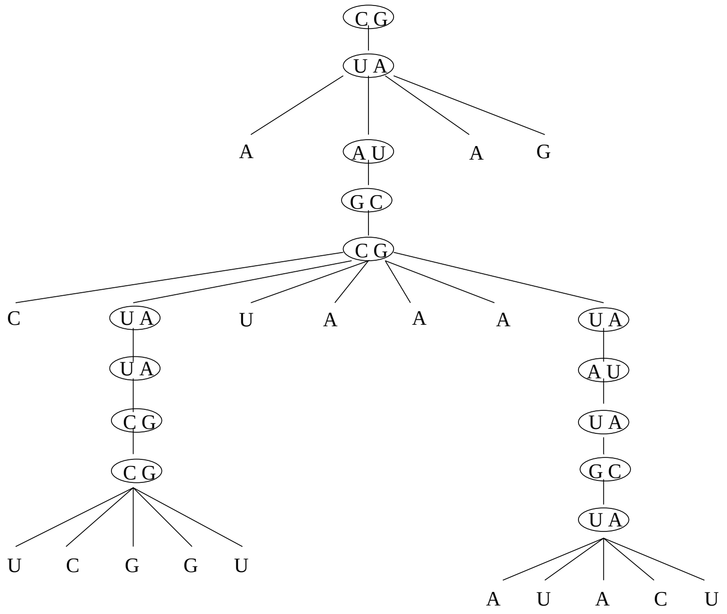

Figure 2: Tree representation of the RNA in Figure 1.

We can use ordered tree edit distance algorithm [14] to determine the minimum number of weighted operations that can change one RNA into another. Only a distance measure may not be very useful of the comparison of RNAs. A weighted mapping between two RNA secondary structures where base-pairs map to base-pairs and unpaired bases map to unpaired bases is more intuitive and useful. Note that the resulting mapping is usually a tree. We denote this version as weighted largest common sub-structure (WLCS) problem. If we assign a nonnegative score for each mapped pair, we can again use [14] for maximum mapping to solve WLCS problem.

This algorithm has a time complexity of $O(|T_1||T_2| \min(depth(T_1), leaves(T_1)) \min(depth(T_2), leaves(T_2)))$ and space complexity of $O(|T_1||T_2|)$ where $|T_i|$ is the size of the tree T_i. The depth is really the collapsed depth where nodes with degree one are ignored when counting the depth.

Using our tree representation, the size of the tree is smaller than the length of the primary structure but they are in the same order. The collapsed depth here is really the maximum number of loops on a path from root to a leaf. Here the loops are bulge loops, internal loops, multiple loops, and hairpin loops. We denote this by dp. Note that this factor dp does not really depend on the length of primary structure or the number of base-pairings (size of S). It only depends on the number of loops in the secondary structure. Therefore this factor is considerably smaller than the size of the RNA sequence.

Taking this into account, our resulting algorithm for comparing two RNA secondary structures with edit distance measure or WLCS will have time complexity of $O(n_1 n_2 dp_1 dp_2)$ and space complexity of $O(n_1 n_2)$, where n_1 and n_2 are the sizes of the RNA primary structures. Using the same algorithm we can also solve WLCS problem. The only thing we need to do is to assign a nonnegative score for each relabel and a zero for each insert and delete, and change minimum to maximum in the algorithm. The time complexity compared favorably with $O(n_1^2 n_2^2)$ in [1] which solves a similar problem.

We can also consider the tree alignment distance where we seek the minimum costs to align two trees. Using the result in [3], we have an algorithm for this problem with time complexity of $O(|T_1||T_2|(degree(T_1)+degree(T_2))^2)$ and space complexity of $O(|T_1||T_2|(degree(T_1) + degree(T_2)))$. Using this algorithm carefully we can ganrantee that the resulting super-tree structure represents an RNA. Using this algorithm with the same scoring scheme

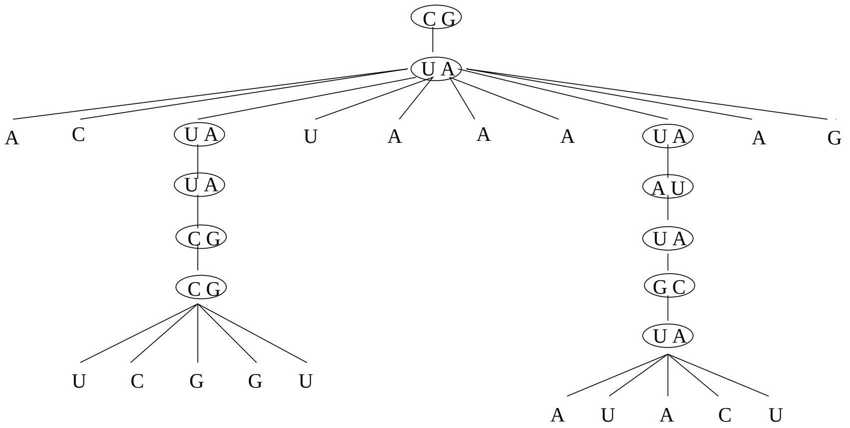

Figure 3: Tree representation of the RNA in Figure 1 after three deletions.

for WLCS, we can solve the WSCS problem. Denote the degree of trees by dg, the time complexity is $O(n_1 n_2 (dg_1 + dg_2)^2)$. Again dg is much smaller than the size of the RNA sequence. The time complexity compared favorably with $O(n_1^2 n_2^2 (n_1^2 + n_2^2))$ in [1] which solves a similar problem.

3 Inferring RNA structure via WLCS and WSCS

We now consider the problem of predicting RNA secondary structure. The input are two RNAs with the first one has both primary and secondary structures, and the second one only has a primary structure. Our goal is to find a secondary structure for the second RNA such that the WLCS (or WSCS) score is maximized.

Let T be the tree representation of the RNA with known secondary structure and R be the RNA with only the primary structure throughout this section. The nodes in the ordered tree T are numbered 1 through $|T|$ according to the *postorder*. Denote the label of node i in R as $r[i]$, and the label of node i in tree T as $l[i]$ and the subtree of T rooted at node i as $T[i]$.

We have two score functions $\gamma(u,v) \geq 0$ where $u, v \in \sum$, and $\delta(u,v) \geq 0$ where $u, v \in \sum \times \sum$. Denote the maximum score between a forest F in T and an interval (k,l) in R as $D(F,(k,l))$. Here the maximum is taking between F and all the possible secondary structures on (k,l) with WLCS measure.

In the following, let i be a node of T with degree d_i. Denote the children of i as $i_1, \ldots, i_{d_i}$. For any s, $1 \leq s \leq d_i$, let $F[i_1, i_s]$ represent the forest consisting of the subtrees $T[i_1], \ldots, T[i_s]$. For convenience, $F[i_1, i_{d_i}]$ is also denoted $F[i]$.

3.1 Properties of the maximum score

The following lemmas form the basis of our algorithm. The first two lemmas are trivial.

Lemma 1

$$\begin{aligned} D(\theta,\theta) &= 0 \\ D(F[i],\theta) &= 0 \\ D(\theta,(k,l)) &= 0 \end{aligned}$$

Lemma 2 *If $l[i]$ is an unpaired base, then*

$$D(T[i],(k,l)) = \max \begin{cases} \gamma(l[i], r[l]) \\ D(T[i],(k,l-1)) \end{cases}$$

Proof: $\square$

Lemma 3 *If $l[i]$ is a base-pair, then*
$D(T[i],(k,l)) =$

$$\max \begin{cases} D(T[i],(k+1,l)) \\ D(T[i],(k,l-1)) \\ D(F[i],(k,l)) \\ D(F[i],(k+1,l-1)) + \delta(l[i],(r[k],r[l])) \\ \quad \textit{if } r[k] \textit{ and } r[l] \textit{ are complementary} \end{cases}$$

Proof: (sketch) Consider the mapping between $T[i]$ and the best secondary structure on (k,l), there are four cases. The first two cases are either $r[k]$ is not mapped or $r[l]$ is not mapped. The third case is that $l[i]$ is not mapped. The last case is that $l[i]$ is mapped

```
begin

  for i := 1 to |T|
    for intervals (k, l), 1 ≤ k < l ≤ n_2
    (* assume that intervals are processed in lexicographically
       increasing order of width *)

      for s := 1 to d_i
        for h := k to l
          Compute D(F[i_1, i_s], (k, h))
          as in Lemma 4 and 5

      compute D[T[i], (k, l))
      as in Lemma 2 and 3

end
```

Figure 4: Algorithm: Inferring Secondary Structure

and both $r[k]$ and $r[l]$ are mapped. Now $r[k]$ and $r[l]$ must be a base-pair and this base-pair $(r[k], r[l])$ mapped to $l[i]$. We find the maximum amount these cases. □

This lemma implies that $D(F[i], (k, l))$ is required for computing $D(T[i], (k, l))$.

Lemma 4 *For any s such that $1 \leq s \leq d_i$, if $l[i_s]$ is an unpaired base, then*

$$D(F[i_1, i_s], (k, l)) = \max \begin{cases} D(F[i_1, i_{s-1}], (k, l)) \\ D(F[i_1, i_s], (k, l-1)) \\ D(F[i_1, i_{s-1}], (k, l-1) + \gamma(l[i_s], r[l]) \end{cases}$$

Proof: (sketch) In this case, $l[i_s]$ is a leaf. Either $l[i_s]$ does not mapped to anything, or $r[l]$ does not mapped to anything, or both are mapped and they have to map to each other. □

Lemma 5 *For any s such that $1 \leq s \leq d_i$, if $l[i_s]$ is a base-pair, then*

$$D(F[i_1, i_s], (k, l)) = \max \begin{cases} D(F[i_1, i_{s-1}], (k, l)) \\ \max_{k \leq h \leq l}\{D(F[i_1, i_{s-1}], (k, h-1)) \\ \qquad +D(T[i_s], (h, l))\} \end{cases}$$

Proof: (sketch) In this case, either the whole tree $T[i_s]$ does not mapped to anything or it mapped to a partial secondary structure generated from interval (h, l) for some h. □

3.2 Algorithm and complexity

Algorithm

These lemmas give us an algorithm solving the RNA inferring problem. The algorithm is given in Figure 4. Once $D(T, (1, n_2))$ is found, one can generate the largest common sub-structure between T and R through backtracking. The based pairs of R in the common sub-structure form the secondary structure of R.

Complexity

Recall that d_i is the number of children of node $t[i]$. Let db_i denote the number of base-paired children of of node $t[i]$ if $d_i > 1$. Otherwise $db_i = 0$. For each node $T[i]$, there are d_i forests of the form $F[i_1, i_s]$. The algorithm considered each forest with each interval. By Lemma 5 only for db_i forests we need to spend $O(n_2)$ time. Note that we do not need to use Lemma 5 when $d_i = 1$ since $F[i]$ is a tree. All the other forests need $O(1)$ time by Lemma 4. Therefore the time complexity is $O(\sum_{i=1}^{|T|}(d_i n_2^2 + db_i n_2^3)) = O(\sum_{i=1}^{|T|} d_i n_2^2 + \sum_{i=1}^{|T|} db_i n_2^3) = O(n_2^2 \sum_{i=1}^{|T|} d_i + n_2^3 \sum_{i=1}^{|T|} db_i) = O(n_1 n_2^2 + lp n_2^3)$. Note that $lp = \sum_{i=1}^{|T|} db_i$ is the number of loops in T which is much smaller than n_1. This is better that $O(n_1^2 n_2^2 + n_1 n_2^3)$ [1].

3.3 Inferring via WSCS

One can also consider the same problem using WSCS measure. We now show that the solution given above by using WLCS is in fact a solution for WSCS.

Lemma 6 *Inferring through WSCS is equivalent to inferring through WLCS.*

Proof: (sketch)

- All internal nodes of an RNA tree representation are base-pairs. All leaf nodes of an RNA tree representation are unpaired bases.
- The nodes in R which are not in the LCS with T are all leaves.
- If delete all these unmatched leaves, we have an alignment of the resulting tree with T.
- We can add all these unmatched leaves back while maintaining the alignment.

This means that the optimal solution of WLCS is also a solution of WSCS. Since any solution of WSCS is also a solution of WLCS, inferring through WSCS is equivalent with inferring through WLCS. $\square$

4 Conclusion

We have presented a tree representation for RNA secondary structures. We have considered several versions of similarity measure between RNA secondary structures and provided efficient algorithms. We also considered the RNA inferring problem base on WLCS and WSCS and presented a good solution.

References

[1] V. Bafna, S. Muthukrishnan, and R. Ravi, 'Comparing similarity between RNA strings', Proc. Combinatorial Pattern Matching Conf. 95, LNCS 937, pp.1-14, 1995

[2] F. Corpet and B. Michot, 'RNAlign program: alignment of RNA sequences using both primary and secondary structures', *Comput. Appl. Biosci* vol. 10, no. 4, pp.389-399, 1995

[3] T. Jiang, L.Wang, and K. Zhang, 'Alignment of trees – an alternative to tree edit', *Theoretical Computer Science* no. 143, pp.137-148, 1995

[4] S.Y. Le, R. Nussinov and J.V. Mazel, 'Tree graphs of RNA secondary structures and their comparisons' *Comput. Biomed. Res.* vol. 22, pp.461-473, 1989

[5] 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* vol. 5, pp.205-210, 1989

[6] R. Nussinov, G. Pieczenik, J.R. Griggs, and D.J. Kleitman, 'Algorithms for loop matchings', *SIAM J. Appl. Math.* 35, pp.68-82, 1978

[7] S.E. Needleman and C.D. Wunsch, 'A general method applicable to the search for similarities in the amino-acid sequences of two proteins', *J. Mol. Bio.*, 48, pp.443-453, 1970

[8] B. Shapiro, 'An algorithm for comparing multiple RNA secondary structures', *Comput. Appl. Biosci* vol. 4, no. 3, pp.387-393, 1988

[9] B. Shapiro and K. Zhang, 'Comparing multiple RNA secondary structures using tree comparisons', *Comput. Appl. Biosci* vol. 6, no.4, pp.309-318, 1990

[10] T.F. Smith and M.S. Waterman, 'The identification of common molecular subsequences', *J. Mol. Bio.* 147, pp.195-197, 1981

[11] T.F. Smith and M.S. Waterman, 'Comparison of biosequences', *Adv. in Appl. Math.* 2, pp.482-489, 1981

[12] K.C. Tai, 'The tree to tree correction problem', *JACM* vol.26, no.3, pp.422-433, 1979

[13] M.S. Waterman and T.F. Smith, 'RNA secondary structure: a complete mathematical analysis', *Math. Biosci.* 42, pp.257-266, 1978

[14] K. Zhang and D. Shasha, 'Simple fast algorithms for the editing distance between trees and related problems', *SIAM J. Computing* vol. 18, no. 6, pp.1245-1262, 1989

[15] M. Zuker, 'On finding all suboptimal foldings from of an RNA molecule', *Science* 244, pp.48-52, 1989

[16] M. Zuker and D. Sankoff, 'RNA secondary structure and their prediction', *Bull. Math. Biol.* 46, pp.591-621, 1984

[17] M. Zuker and P. Stiegler, 'Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information', *Nucleic Acid Res.* 9, pp.133-148, 1981