A Rule-based Approach for RNA Pseudoknot Prediction

X.Z. Fu[1], H.Wang[1], W. Harrison[3], and R.Harrison[1,2]
[1]*Department of Computer Science and* [2]*Department of Biology,*
Georgia State University, Atlanta GA 30303, USA
[3]*Department of Computer Science*
University of Missouri, Columbia MO 65211, USA

Abstract

RNA plays a critical role in mediating every step of cellular information transfer from genes to functional proteins. *Pseudoknots* are functionally important and widely occurring structural motifs found in all types of RNA. Therefore predicting their structures is an important problem. In this paper, we present a new RNA pseudoknot structure prediction method based on term rewriting. The method is implemented using the Mfold RNA/DNA folding package and the term rewriting language Maude. In our method, RNA structures are treated as terms and rules are discovered for predicting pseudoknots. Our method was tested on 211 pseudoknots in PseudoBase and achieves an average accuracy of 74.085% compared to the experimentally determined structure. In fact, most pseudoknots discovered by our method achieve an accuracy of above 90%. These results indicate that term rewriting has a broad potential in RNA applications ranging from prediction of pseudoknots to discovery of higher level RNA structures involving complex RNA tertiary interactions.

1. INTRODUCTION

It is now well recognized that RNA structure may be described with ideas from formal language theory (e.g. context-free grammars). Primary RNA structures are simply

strings of nucleotides and many researchers have applied string-based algorithms and techniques to the problem of RNA structure determination. This paper applies another idea from the study of languages – *term rewriting* (Baader and Nipkow, 1999) – to structure prediction.

Term rewriting is a style of computation in which an input – the term – is transformed according to a predetermined set of rules. Term rewriting has a long history in theoretical Computer Science (Baader and Nipkow, 1999) and has recently found a place in bioinformatics applications (Eker et al., 2002; Talcott et al., 2004) as well. The method described in this paper treats RNA structures as terms and discovers rules for predicting pseudoknots.

1.1 RNA Structure and Pseudoknots

RNA primary structure is the nucleotide sequence of four bases A (adenine), C (cytosine), G (guanine), U (uracil). The pattern of base pairing determines the secondary structure of RNA. Watson-Crick (A=U and G≡C) and Wobble (G=U) are widely occurring stable base pairs in RNA, while other less stable base pairs are possible but often ignored. The secondary structure can be decomposed into a few types of secondary structural motifs: stem, hairpin loop, bulge, internal loop, multi-branched loop, start sequence and external sequence (Batey et al., 1999), which are shown in Figure 1.

From the computational viewpoint, the challenge of the RNA structure prediction problem arises from some special structures called *pseudoknots*, which are defined as follows. Let S be an RNA sequence $S = a_1 a_2 a_3 a_4 \ldots a_n$ and (a_i, a_j) and (a_h, a_k) be two distinct

base pairs in the sequence, where $i < j$ and $h < k$. A pseudoknot is composed of two interleaving base pairs such that $1 \le i < h < j < k \le n$ or $1 \le h < i < k < j \le n$.

1.2 Methods for Pseudoknot Prediction

Prediction of RNA structure with pseudoknots is inherently challenging. It has been demonstrated that the prediction of pseudoknots within RNA structure is an NP-complete problem when using free energy minimization methods (Lyngso and Pedersen 2000). Current approaches to pseudoknot prediction mainly fall into three categories: comparative sequence analysis (CSA) (Witwer et al., 2004; Tabaska et al., 1998; Wuyts et al., 2000; Ruan et al., 2004), energy minimization through polynomial dynamic programming (DP) (Deogun et al., 2004; Ruan et al., 2004), and heuristic search-based methods (Batenburg et al., 1995; Gultyaev et al., 1995).

CSA efficiently predicts consensus structures when sufficiently large sets of homologous RNA sequences are available. However, determining the consensus structure through comparative analysis requires a good alignment of the homologous sequences and such sequences are not always available.

DP is successful when restricted to relatively short pseudoknots and does not require homologous sequences. However, DP algorithms are impractical for pseudoknots of length beyond several hundred nucleic bases as a result of their inherent computational complexity.

One method of coping with the computational complexity of structure prediction is to adopt heuristic search techniques such as Monte-Carlo simulation and genetic algorithms; however, such techniques neither guarantee the discovery of optimal structures nor do they predict the distance from an optimal solution.

Matsui et al. (2005) recently proposed a method based on pair stochastic tree adjoining grammars for aligning RNA secondary structure including pseudoknots. They predicted the structure of an RNA sequence by aligning the sequence into a 'folded' skeletal tree which is parsed from certain known pseudoknot structures. The dependency between the prediction results and the selected RNA with known structure in this method does not exist in our method because our method uses only single sequences for prediction without considering any other known RNA structures.

1.3 Motivation

The formation of RNA tertiary structures is primarily dominated by three varieties of interactions (Batey et al., 1999; Pley et al., 1994; Cate et al., 1996; Ferre-D'Amare et al., 1998). These interactions occur between (i) two double-strand helical regions, (ii) one double-strand helical region and an unpaired region, or (iii) two unpaired regions. Here, an unpaired region refers to a hairpin loop, internal loop, bulge, multi-branched loop, start sequence, and external sequence (see Figure 1). Our current approach focuses on the interaction of two unpaired regions; however, we believe our method extends to other interaction types. We leave these questions for future work.

The current method explores the prediction of RNA pseudoknots based on secondary structures by considering the interactions between unpaired regions. This approach is motivated largely by two observations about pseudoknots. The first of these is that the interaction between two unpaired regions gives rise to pseudoknots and the second is that algorithms for pseudoknot-free structure prediction are sufficiently accurate and quick even in cases including a large number of nucleotides (Mathews et al., 1999).

Our method has four steps (see Figure 2). In the first step, the RNA secondary structure is predicted from RNA sequence. Step 2 parses the secondary structure using term rewriting to retrieve motifs. Step 3 performs motif-motif interactions by certain rules and a score function is applied to evaluate each motif-motif interaction. Step 4 outputs the predicted structure.

2. PSEUDOKNOT PREDICTION BASED USING TERM REWRITING

Our method for pseudoknot prediction is presented below in Section 2.2. Before proceeding, however, an overview of term-rewriting in Maude is given. This overview is necessarily brief and readers requiring more explanation should consult the references (particularly, Clavel et al., 2003).

2.1. Preliminary

- **Term rewriting and Maude**

Maude, as a term rewriting language, supports both equational and rewriting logic computation for a wide range of applications with high performance (Clavel et al., 2003). The time cost of our calculation is dominated by *Mfold* (Zuker 2003) calculation of secondary structure.

- **Terms in Maude:**
  - *sort*: sort can be considered as a type of collection. *subsort* term can be used to indicate a belonging relationship between sorts.
  - *op*: is used to define an operator. It enables the user-definable syntax in Maude. Operator declarations may include attributes that provide additional information about the operator, like *associativity*, *commutativity* et al.

- *eq:* stands for equation. It demonstrates a bidirectional equivalent relationship between two sorts. Equation can be used to deploy reduction and conversion in rewriting logic language by defined rules. *ceq* declares a conditional equation.
- *rl:* defines rewrite rules for dynamic behaviors. Computationally, rewrite rules specify local concurrent transitions. Unlike equations, rewrite rules are irreversible. *crl* is used to define conditional rules.

- ***Mfold***

Mfold is one of the most widely used software package for RNA/DNA secondary structure prediction based on free energy minimization (Zuker 2003). We use *Mfold* version3 in our implementation.

2.2. Method

Here we explain our model step by step by giving an example of Viral 3'-UTR RNA pseudoknot (PKB116)(Batenburg et al., 2000). Figure 3 shows the prediction process. In Figure 3, the left part is our method and the right part is the results corresponding to each step of the method.

Step 1:
The step 1 generates pseudoknot-free secondary structure from an RNA sequence. In our practice, *Mfold* package is used with default parameters. The output secondary structure of step 1 is a dot-bracket string in which corresponding brackets stand for base pairs of nucleic bases.

```
.........[[[[[ ... [[.....]].]]]]].
```

Step 2:
Step 2 retrieves secondary structural motifs (see Figure 1) from the dot-bracket string. Here, multi-branched loop is treated as independent internal loops.

The motifs in the example of Figure 3 are as follows:

```
S(CAGUGUUUU)T(GAAGU)I(CCA)T(CU)H(UAAAU)T(AG)I(A)T(ACUUC)E(U)
```

where S(CAGUGUUUU) is a start sequence; T(GAAGU), T(CU), T(AG), and T(ACUUC) are stems; I(CCA) and I(A) are internal loops; H(UAAAU) is a hairpin loop; E(U) is an external sequence.

Additional modifications on the stems are necessary for pseudoknot prediction because nucleic bases in a stem may be involved in the pseudoknot. Hence, the base pairs in a stem whose length is less than a predefined value will be separated. After separating certain stems, motifs need to be parsed again. It is noticeable that the bases pairs in stems T(CU) and T(AG) are separated. Now the dot-bracket string is:

```
.........[[[[[............]]]]].
```

Parsing this string, we get motifs:

```
S(CAGUGUUUU)T(GAAGU)H(CCACUUAAAUAGA)T(ACUUC)E(U)
```

Definition:

- *' and '*: to facilitate retrieving motifs from the dot-bracket string, we add *' and '* symbols into the dot-bracket string to label the beginning and ending of the string.
- *sorts:* sorts defined in our model.(see Figure 4, Figure 5)

In this step, motifs are retrieved as follows:

- Find start motif and external motif, i.e. S(), E().

The S motif has a pattern that it must begin with a start symbol (‘), followed by one or more dot (.) and ended with a left bracket ([ ). This pattern can be outlined by the following Maude code:

```
1  op .:→DotSet .
2  ops [ ]:→BracketSet .
3  op __:DotSet DotSet→DetSet .
4  op ‘_[:DotSet→MotifSet .
5  vars D D1:DotSet .
6  eq ‘D[=S(D)[ .
```

The pattern of E motif is that it must begin with a right bracket ( ]), followed by one or more dot (.) and ended with a end symbol (’). It can be deployed in Maude as:

```
7  op ]_’:DotSet→MotifSet .
8  eq ]D’=]E(D) .
```

- Find hairpin loop motif, i.e. H().

The hairpin loop motif begins with a left bracket ([ ), followed by one or more dot (.) and ended with a right bracket ( ]).

```
9   op [_]:DotSet→MotifSet .
10 eq [D]=[H(D)] .
```

- Find bulge motif and internal loop motif, i.e. B(),I().

These two motifs have something in common. They must contain a part having a pattern as either [. $^{(m)}$ [or] . $^{(m)}$] (excluding multi-branched case, which we will discuss below), where $m \geq 1$. For a bulge loop motif, a right scans of [. $^{(m)}$ [part will encounter the first right bracket immediately followed by another right bracket, whereas an internal loop motif will see the first right bracket directly followed by something but a right bracket symbol. The difference distinguishes a bulge motif from an internal loop motif. The same thing holds for a] . $^{(m)}$] part if some direction changes are taken.

Hence, the Maude code for bulge motif is:

```
11  var M M1:MotifSet .
12   op [_[_]]:DotSet MotifSet→MotifSet .
13   eq [D[M]]=L([)B(D)L([)MR(])R(]) .
```

And the code for internal loop motif is:

```
14  op [_[_]_:DotSet MotifSet DotSet→MotifSet .
15  eq [D[M]D1=[I(DL([)MR(])D1 .
16  op [_[_]_:DotSet MotifSet MotifSet→MotifSet .
17  eq [D[M]M1=[I(DL([)MR(])M1 .
```

The code for] . $^{(m)}$] part is skipped here. An internal loop motif in multi-branched introduce another pattern which is:] .$^{(m)}$ [. It can be recognized by the following code:

```
18   op _]_[_:MotifSet DotSet MotifSet→MotifSet .
19   eq M]D[M1=M]I(D)[M1 .
```

• Other reduction steps for the parsing

Besides parsing individual motif, more operators and equations are necessary to reduce brackets such that the dot-bracket string can be correctly parsed into motifs we need.

(1) Bracket reduction

Nested continuous pairs of brackets can be described in a pattern like [[MotifSet]]. The inner pair of brackets will not have any impact on the motif determination. Thus, they can be reduced.

```
20  op [[_]]:MotifSet →MotifSet .
21  eq [[M]] =[L()MR()] .
```

Similarly, the brackets in a pattern like '[MotifSet]'can be reduced as follows:

```
22  op '[_] ':MotifSet →MotifSet .
23  eq '[M] ' ='L()MR()' .
```

(2) Stem Motif concatenation

Nested continuous pairs of stem motifs can be further concatenated if they fall into certain pattern like L()L()MotifSetR()R(). The Maude code is:

```
24 op _ _ _ _ _:MotifSet MotifSet MotifSet MotifSet MotifSet →MotifSet .
25  eq L(loop)L(loop1)MR(loop2)R(loop3)=L(looploop1)MR(loop2loop3) .
26  op _ _ :MotifSet MotifSet→MotifSet (commu) .
```

Finally, convert all L() and R() motifs into T() motifs using the following code:

```
27 var t : StemSet .
28 eq t( )M = T()M .
```

The code from line1 to line28 parses the dot-bracket string into motifs. The code for re-parsing the dot-bracket string after separating stems is skipped here. Finally we get the motifs used in next step by removing all stems:

```
S(CAGUGUUUU)H(CCACUUAAAUAGA)E(U)
```

Step 3:

This Step performs *permissible* motif-motif interaction based on our rules. A score function is applied to evaluate each motif-motif interaction.

- ***Permissible*** **motif-motif interactions**

We enforce motif-motif interactions by the following rules;

(a) A motif never interacts with itself.

(b) Each H or B motif can interact with all other motifs.

(c) Each I or S motif can interact with the motifs behind itself towards the end of the structure until encounters an H motif.

(d) The E motif can interact with the motifs before itself towards the beginning of the structure until encounters an H motif.

- **Score function**

Each motif-motif interaction is scored by taking the weight of base pair region and the distance penalty of the base pair region into consideration. The weights in the score function are chosen to reflect the physical chemistry of nucleic acids (Bloomfield et al., 1999). A base pair region is defined as:

$region = \{(i, j),(i-1, j+1),...(i-m, j+m)\}$, where $i \in motif1, j \in motif2, i < j, m = length(region)$ and each base pair in this region belongs to the set of $\{CG, GC, AU, UA, GU, UG\}$.

The weight of *region* is defined as: $W_{region} = \sum_{(i,j)\in region} weight(i, j)$,

Where weight(CG/GC):weight(AU/UA):weight(GU/UG) =3:2:1. The weights are approximate values that indicate the trends in base pair energy.

The distance penalty of the base pair region is defined as Dis_{region}= i-j, where (i, j) is the closest base pair in the region and i<j. Then, score function can be defined as:

$Score_{motif-motif} = Max(\alpha \times W_{region} + (1-\alpha) \times Dis_{region})$ where α is a constant and $\alpha \in [0,1]$. The constant α is a heuristic parameter adjusting the significance of W_{region} and Dis_{region} in the score function. In our experiment, α is set to 0.8.

- **Format of motifs**

A motif has a format like: $MotifType(seq : String, max : Int)$, where MotifType∈{H,I,B,S,E}; *seq* is a variable storing the nucleotide sequence of a motif; *max* is a variable storing the maximal score of each motif when it interacts with other motifs. Each motif has an initial max score 0. The motifs we will deal with are:

```
S(CAGUGUUUU,0)H(CCACUUAAAUAGA,0)E(U,0)
```

Figure 5 shows the sorts defined in Step3.

Maude code for motif-motif interaction is as follows:

***(a) Operators and variables

```
1  op _(_,_):Hairpin String Int→HMotif .
2  op _(_,_):NonHairPin String Int→NonHMotif .
3  op _ _:NonHMotifSet NonHMotif→NonHMotifSet .
4  op _ _: MotifSet MotifSet→MotifSet .
5  op _ _ _:Motif MotifSet Motif→MotifSet .
6  op _ _ _:NonHMotif NonHMotifSet NonHMotif→NonHMotifSet .
```

Line 1 defines the operator to recognize hairpin loop motifs. Line 2 defines the operator to recognize non-hairpin loop motifs. Line 3 to line 6 define the operators which deal with multiple continuous motifs.

```
7   vars seq seq1:String .                  ***sequence of a motif
8   vars max max1:Int .                     *** maximal score
9   vars mSet:MotifSet NonHSet:NonHMotifSet .
10  var M:MotifType .                       ***M is H,I,S,E, or B
11  var Z:NonHairpin .                      ***Z is I,S,E, or B
12  var X:HBtype .                          ***X is H or B
13  var Y:IStype .                          ***Y is I or S
```

Line 7 to line 13 define variables used by lines 14-21.

***(b) H and B motifs

```
14  ceq X(seq,max)M(seq1,max1)=X(seq,MAX(seq,seq1))M(seq1,max1)
                                  if MAX(seq,seq1)>max .
15  ceq X(seq,max)mSetM(seq1,max1)=X(seq,MAX(seq,seq1))mSetM(seq1,max1)
                                  if MAX(seq,seq1)>max .
```

Line 14 and line 15 let the X motif interact with the motifs towards to the end of the structure.

```
16  ceq M(seq1,max1)X(seq,max)=M(seq1,max1)X(seq,MAX(seq,seq1))
                                  if MAX(seq,seq1)>max .

17  ceq M(seq1,max1)mSetX(seq,max)=M(seq1,max1)mSetX(seq,MAX(seq,seq1))
                                  if MAX(seq,seq1)>max .
```

Line 16 and line 17 let the X motif interact with the motifs towards to the beginning of the structure.

***(c) I and S motifs

```
18  ceq Y(seq,max)Z(seq1,max1)=Y(seq,MAX(seq,seq1))Z(seq1,max1)
                                  if MAX(seq,seq1)>max .

19  ceq Y(seq,max)NonHSetZ(seq1,max1)=Y(seq,MAX(seq,seq1)) NonHSet
                                  Z(seq1,max1) if MAX(seq,seq1)>max .
```

Line 18 and line 19 let the Y motif interact with the motifs towards to the end of the structure.

***(d) E motif

```
20  ceq Z(seq1,max1)E(seq,max)= Z(seq1,max1)E(seq,MAX(seq,seq1))
                                  if MAX(seq,seq1)>max .

21  ceq Z(seq1,max1)NonHSetE(seq,max)=
        Z(seq1,max1)NonHSetE(seq,MAX(seq,seq1)) if MAX(seq,seq1)>max .
```

Line 20 and line 21 let the Z motif interact with the motifs towards to the beginning of the structure.

In the above code, MAX(seq:String, seq1:String) is a module which calculates the score of each motif-motif interaction by implementing the score function and seq and seq1 are variables storing the nucleotide sequences of two motifs. Maude provides convenient string processing operators such as *find*, *substr* et al. It is easy to implement the MAX module in Maude.

Finally, the global maximal score of all motif-motif interactions is found. The potential pseudoknot is located between the two motifs with the global maximal score.

Step 4:

The motifs with the highest score are considered as the candidates forming the potential pesudoknot. The stems separated in step 2 may or may not be recovered before outputting the final structure. The effect of this recovery operation will be discussed in the data analysis section. The pseudoknot in the final output structure is labeled with '{' and '}', as the example output in the Figure 3.

```
.{{{{....[[[[[.}}}}.........]]]]].
```

Figure 6 shows three structures of the example. (A) is the secondary structure predicted by *Mfold*, (B) is our prediction, and (C) the experimental structure. Comparing our prediction with the experimental structure, the accuracy we get is 93.939%.

3. DATA ANALYSIS

3.1 Evaluation Criteria

For each nucleic base a_i in the sequence, assume a_j and a_k are the partners of a_i in the predicted structure and reference structure respectively with $0 \le j,k \le n$ and $1 \le i \le n$. If the partner of a_i is a_0, a_i is unpaired. For a_i, our prediction has two possible results:

(1) Correct if $j = k$ (2) Wrong if $j \ne k$

Accuracy is defined as follows: $Accuracy = 100\% \times \frac{\#Correct}{n}$

3.2 Data Analysis

Our model was tested on 211 single-strand pseudoknots in *PseudoBase* (Batenburg et al., 2000) with length varies from 21 to 137. These pseudoknots are classified into 13 groups by *PseudoBase* as shown in Table 2. In Step 2, the bases of the stems are separated according to the length of stems. To specify the effect of stems on pseudoknots, we test the 211 pseudoknots by adjusting two parameters in our method*: stem-length(L)* and *recovery*. If a stem with a length less than L, the base pairs in the stem will be separated. When *recovery* is 'yes', any stem separated in step 2 will be recovered if no base in this stem is involved in pseudoknot. From the Table 1, we can see that each case (a)-(d) has higher accuracy than case (e) in which no stem is separated. This indicates that small stems have effect on pseudoknots. Comparing case (a) with (b) and case (c) with (d), we can see that simply recovering stems based on the secondary structure does not contribute to the pseudoknot prediction. The highest accuracy is obtained in (d). Other combinations of the two parameters are tested but cannot result in as good prediction as that when L is 3 or 4 and *recovery* is 'no'.

Testing the 211 pseudoknots using parameters that *stem-length* is 4 and *recovery* is 'no', our model achieves an average accuracy of 74.085%. 36 pseudoknots reach 100% accuracy. Only six pseudoknots have accuracy lower than 30%. The Figure 7 shows the accuracy distribution of 211 pseudoknots.

We compare our method with the algorithm introduced in (Ruan et al., 2004). Their method was implemented in their web server (http://cic.cs.wustl.edu/RNA/) which supports thermodynamic and comparative analysis for prediction of RNA secondary structure with pseudoknots. We tested the 211 pseudoknots on this web server by using their default parameters. The result is described in Table 2.

In Table 2, there are 211 pseudoknots in 13 groups. The accuracies of groups 1, 3, 5, 6, and 9 are more indicative than other groups because they cover most of the pseudoknots in the *PseudoBase*. From Table 2, we can see that our method obtains much higher average accuracy than both *Mfold* and Ruan's server. *Mfold* is specially designed to predict RNA pseudoknot-free secondary structures, and the base pair accuracy is therefore an example of a random expectation.

4. CONCLUSION

Exploring the rules and patterns of RNA structure requires logic programming while constrained by limited knowledge. We have shown how to efficiently use *Maude* for RNA pseudoknot prediction. One can easily add more rules into the model without increasing computational complexity and coding effort, thus improving the extensibility and maintainability. Our experiments provide further evidence that term rewriting is an effective and flexible foundation for building and analyzing complex biological problems, defining new data and rules, and executing reduction and queries using logical

derivation. We are currently exploring new rules based on term rewriting to predict higher-level RNA structures involving complex tertiary interactions.

Acknowledgements:

This work was partially supported by NIH1 P20 GM065762-01A1, the Georgia Research Alliance and the Georgia Cancer Collation. Dr. Robert Harrison is a Georgia Cancer Collation Distinguished Scholar.

References:

Baader, F. and Nipkow, T., (1999) 'Term Rewriting and All That', *Cambridge, England: Cambridge University Press*.

Eker, S., Knapp, M., Laderoute, K., Lincoln, P., and Talcott, C., (2002) 'Pathway Logic: Executable Models of Biological Networks', *Electr. Notes Theor. Comput. Sci*. 71.

Talcott, C., Eker, S., Knapp, M., Lincoln, P., and Laderoute, K., (2004) 'Pathway logic modeling of protein functional domains in signal transduction', *Pac. Symp. Biocomput.* 568-80.

Zuker, M., (2003) 'Mfold web server for nucleic acid folding and hybridization prediction', *Nucleic Acids Res.* 1;31(13):3406-15.

Batey, R.T., Rambo, R.P., and Doudna, J.A., (1999) 'Tertiary Motifs in RNA Structure and Folding', *Angew. Chem. Int. Ed. Engl.* 38(16):2326-2343.

Lyngso, R. and Pedersen, C., (2000) 'RNA pseudoknot prediction in energy-based models', *J. Comput. Biol.*. 7: 409–427.

Witwer, C., Hofacker, I.L., and Stadler, P.F., (2004) 'Prediction of consensus RNA structures including pseudoknots', IEEE*/ACM Trans.Comp.Biol.Bioinf.*,1: 66-77.

Tabaska, J.E., Cary, R.B., Gabow, H.N., and Stormo, G.D., (1998) 'An RNA folding method capable of identifying pseudoknots and base triples', *Bioinformatics*. 14(8):691-9.

Wuyts, J., Rijk, P.D., de Peer, Y.V., Pison, G., Rousseeuw, P. and Wachter, R.D., (2000) 'Comparative analysis of more than 3000 sequences reveals the existence of two pseudoknots in area V4 of eukaryotic small subunit ribosomal RNA', *Nucleic Acids Res.,* 28, 4698–4708.

Deogun, J.S., Donts, R., Komina, O. and Ma, F.,(2004) 'RNA Secondary Structure Prediction with Simple Pseudoknots', *APBC*: 239-246.

Ruan, J.H., Stormo, G.D., Zhang, W.X., (2004) 'An Iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots', *Bioinformatics* 20(1): 58-66.

Batenburg, F.H.van, Gultyaev, A.P., and Pleij, C.W.A., (1995) 'An APL-programmed genetic algorithm for the prediction of RNA secondary structure', *J. Theor. Biol.*.174, 269–280.

Gultyaev, A.P., Batenburg, F.H.van, and Pleij, C.W.A., (1995) 'The computer simulation of RNA folding pathways using a genetic algorithm', *J. Mol. Biol.*. 250:37–51.1995.

Mathews, D.H., Sabina, J., Zuker, M., and Turner, D.H., (1999) 'Expanded Sequence Dependence of Thermodynamic Parameters Improves Prediction of RNA Secondary Structure', *J. Mol. Biol.* 288, 911-940.

Matsui, H., Sato, K., Sakakibara, Y., (2005) 'Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures', *Bioinformatics,* 21(11): 2611-2617.

Pley, H.W., Flaherty, K.M. and McKay, D.B., (1994) 'Three-dimensional structure of a hammerhead ribozyme', *Nature*. 372, 68-74.

Cate, J.H., Gooding, A., Podell, E., Zhou, K., Golden, B., Kundrot, C., Cech, T.R. and Doudna, J.A. (1996) 'Crystal structure of a group I ribozyme domain Principles of RNA packing', *Science* 273, 1678-85.

Ferre-D'Amare, A.R., Zhou, K. and Doudna, J.A. (1998) 'Crystal structure of a hepatitis delta virus ribozyme', *Nature*. 8;395(6702):567-74.

Clavel, M., Durán, F., Eker, F.,Lincoln, P., Martí-Oliet, N., Meseguer, J., and Talcott, C., (2003) 'The Maude 2.0 System', *In Proc. Rewriting Techniques and Applications, Springer-Verlag LNCS 2706,*76-87.

Batenburg, F.H.van, Gultyaev, A.P., Pleij, C.W.A., Ng, J. and Oliehoek, J., (2000) 'Pseudobase: a database with RNA pseudoknots', *Nucl. Acids Res.* 28,1, 201-204.

Bloomfield, V.A., Crothers, D.M., and Tinocco, I., (1999) 'Nucleic acids: structures, properties, and functions', *University Books*, Sausalito, CA.

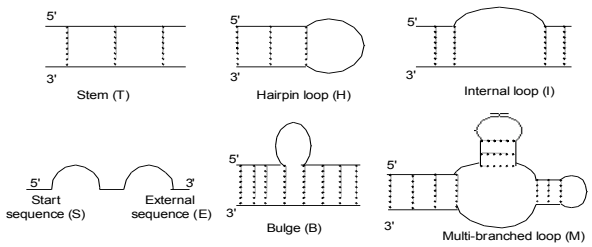

Figure 1. RNA secondary structural motifs.

T: stem, H: hairpin loop, I: internal loop, S: start sequence, E: external sequence, B: bulge, M: multi-branched loop.

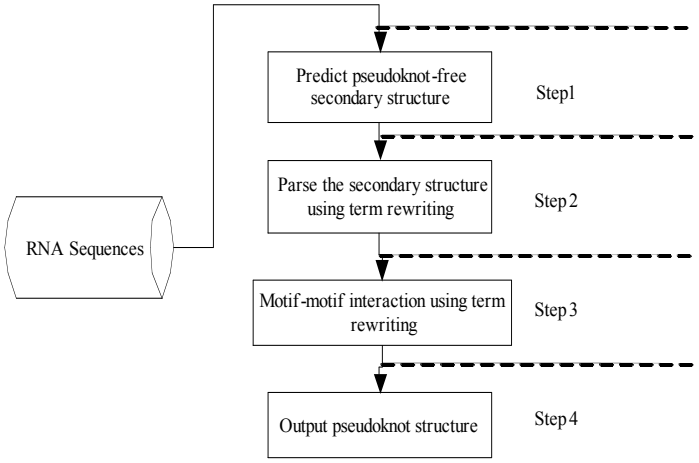

FIGURE 2. METHOD FOR RNA PSEUDOKNOT PREDICTION

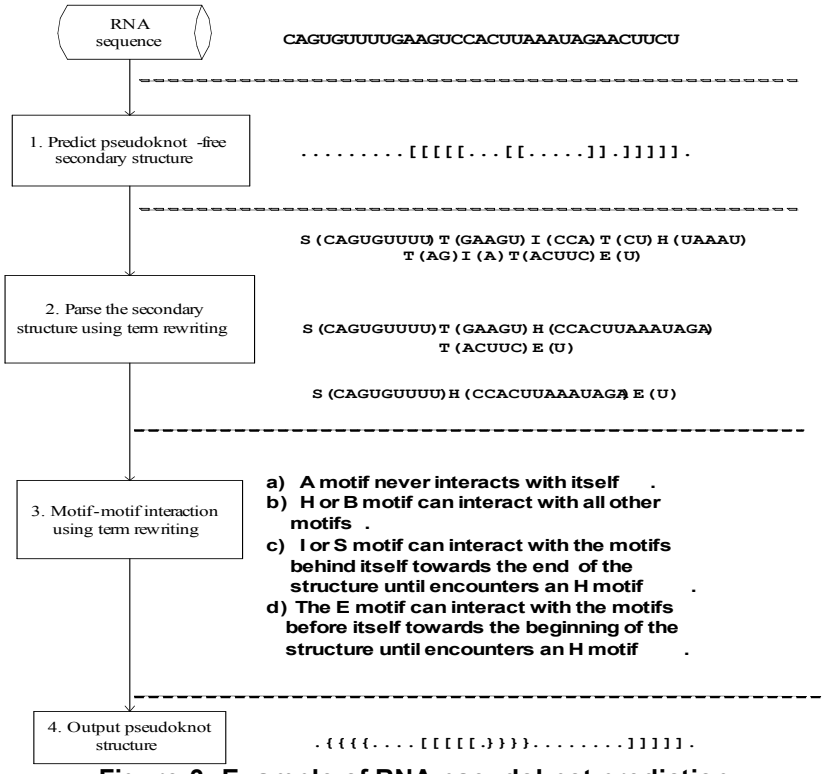

Figure 3. Example of RNA pseudoknot prediction

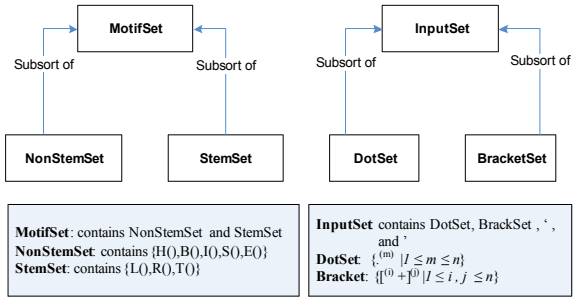

Figure 4. Definition of sorts used in the Step 2

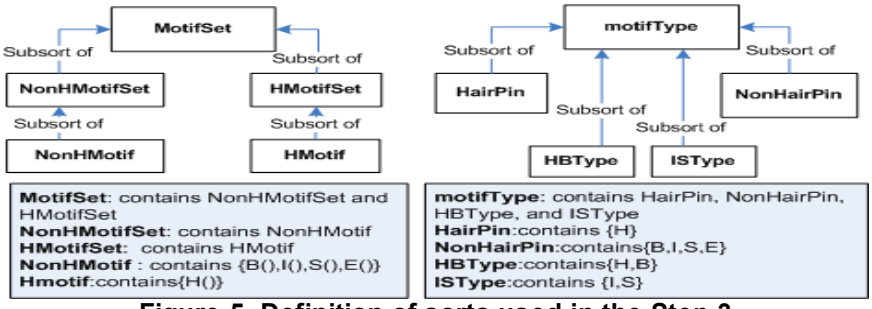

Figure 5. Definition of sorts used in the Step 3

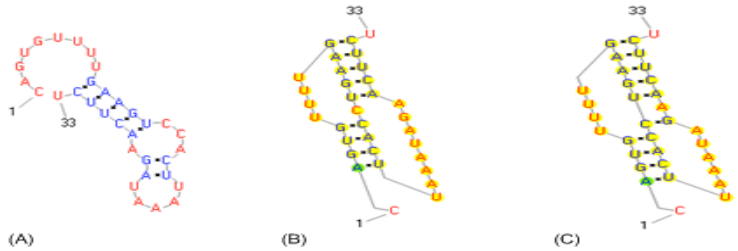

Figure 6. Final output results.
(A) *Mfold* predicted structure (B) Our predicted structure (C) Experimental structure

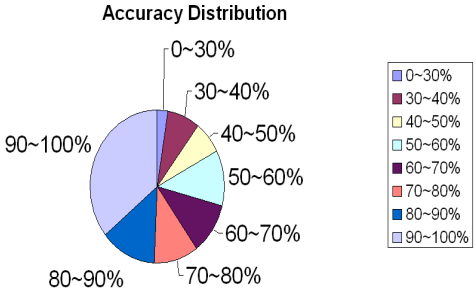

Figure 7. Accuracy Distribution

Table 1. Effect of stem-length and recovery

	Stem-length(L)	*recovery*	Average Accuracy (%)
(a)	3	yes	72.412
(b)	3	no	72.707
(c)	4	yes	73.277
(d)	4	no	74.085
(e)	N/A	N/A	70.878

Table 2. Comparison of accuracy

PseudoBase (Batenburg et al., 2000)	#RNA	Length	Accuracy (%)		
			Mfold (zuker 2003)	Our method	Ruan's server (Ruan et al. 2004)
1. Viral ribosomal frameshifting signals	15	39-73	54.983	63.136	56.317
2. Viral ribosomal readthrough signals	6	61-63	31.388	38.1	51.89
3. Viral tRNA like structures	54	37-137	59.459	63.745	53.014
4. Other viral 5'-UTR	1	35	71.429	100	37.143
5. Other viral 3'-UTR	80	21-96	66.94	88.257	31.822
6. Viral others	20	24-77	65.745	77.13	36.552
7. rRNA	3	46-51	51.161	55.843	36.676
8. mRNA	7	28-120	51.362	60.04	40.66
9. tmRNA	10	30-90	56.316	61.849	36.356
10. Ribozymes	3	73-89	45.02	57.94	36.94
11. Aptamaers	6	33-57	70.976	79.443	40.029
12. Artificial molecules	1	26	79.923	100	38.462
13. Others	4	35-121	65.7155	70.036	46.248
Average Accuracy			***61.6107***	***74.085***	***41.264***