A Matrix Algorithm for RNA Secondary Structure Prediction

S.P.T. Krishnan¹, Mushfique Junayed Khurshid², and Bharadwaj Veeravalli²

¹ Institute for Infocomm Research,
Fusionopolis, Connexis South Tower, #21-01

1 Fusionopolis Way, Singapore 138632

² National University of Singapore,
21 Lower Kent Ridge Road, Singapore 119077

krishnan@i2r.a-star.edu.sg, {mushfiquejk,elebv}@nus.edu.sg

Abstract. In this paper we propose a novel high-performance algorithm, referred to as MARSs (Matrix Algorithm for RNA Secondary Structure Prediction), for predicting RNA Secondary Structures with or without pseudoknots. The algorithm is capable of operating in both serial and parallel modes. The algorithm will take complete advantage of the explicit hardware parallelism increasingly available in today's multi-core processors resulting in execution speedups. Unlike Dynamic Programming based algorithms, MARSs is non-recursive by design and therefore eliminates some of the disadvantages of Dynamic Programming based algorithms. We performed a large-scale experiment on a multi-core hardware using real sequences with verified structures. We detail and discuss the results from these experiments using metrics such as performance gains, run-times and prediction accuracy. This is one of the first attempts of its kind to provide a complete flexibility in evolving a RNA secondary structure with or without pseudoknots using a matrix-based approach.

Keywords: RNA secondary structure prediction, parallel computing, high performance computing, multi-core.

1 Introduction

RNA plays several important roles in a living cell – carries genetic information, acts as catalyst for various cellular processes and also plays a vital role in gene expression. RNA can be represented in one of three forms - primary, secondary and tertiary. The tertiary structure of RNA determines its function and the RNAs *secondary structure* significantly affects the three dimensional shape of the RNA. Therefore, predicting RNA secondary structures is key in determining the 3D structure of a RNA molecule and also in inferencing the RNA's functions and behaviors.

The research community have over several years proposed several RNA secondary structure prediction algorithms. Many current & popular RNA secondary structure prediction algorithms are based on Dynamic Programming derivatives.

Some of the well-known algorithms are as follows - Nussinov's algorithm [1], Waterman's algorithm [2], MFOLD [3], PKNOTS [4], Akutsu's algorithm [5], Dirks' algorithm [6], PKNOTSRG [7], Jitender's algorithm [8] and Ruan's algorithm [9]. Many Dynamic Programming based algorithms are unable to predict pseudoknots and restrict themselves to predicting secondary structures comprising of only simple structural motifs.

In the last 2 decades, the computing industry has been following Moore's law and making faster chips year-on-year basis. Traditionally, the speed gain was obtained by increasing the clock speed of the processor core. However, during the last few years the processor manufacturers have been adding more processing cores instead of making a single core faster due to physical sciences limitations. This requires redesign of current algorithms in many industries. Dynamic Programming algorithms are recursive by design and this inhibits the algorithm from being parallelized on a multi-core system. The primary road block being that later iterations depend on the results from earlier iterations. This requires the design and development of a new class of algorithm that works equal well on single-core and multi-core architectures.

In this paper, we introduce a new algorithm for predicting RNA secondary structures called *MARSs* (Matrix Algorithm for RNA Secondary Structure prediction). *MARSs* does not use the popular dynamic programming methods and therefore we believe it to be the first-of-its-kind. *MARSs* is capable of predicting both Pseudoknots and Non-Pseudoknot structures with equal ease. *MARSs* has been designed with parallelism in mind and can easily scale from single-core to many-cores resulting in significant speedup and with no degradation in the quality of output. We used *MARSs* to predict secondary structures of real RNA sequences and also observed the speedup as a result of using many incremental multi-cores system with each subsequent system have one more processing core than the previous system. We report these results in this paper.

2 RNA Secondary Structures

RNAs are functionally important and play a key role in various cellular processes such as ribosomal frame-shifting, control of translation and splicing [10]. RNA primary structure is the sequence of nucleotide bases that comprise the RNA, namely **A** (adenine), **C** (cytosine), **G** (guanine) and **U** (uracil). Base pairings of these nucleotides are the main determinants of the secondary structures of RNA. Stable base pairs in RNA like *Watson-Crick* (**A-U** and **G-C**) and *Wobble* (**G-U**) are common, but weaker base pairs like *Hogsteen* (**A-C**) are also possible. Moreover, there are some common structural motifs that occur in RNA Secondary structures: Double-stranded segment, Bulges, Symmetric or Asymmetric internal loops, Hairpins, Two-stem junctions (coaxial stacks), Kissing hairpins [11].

In addition to the above simple motifs, *Pseudoknots* are also a common occurrence in RNA. *Pseudoknots* are an important structural element present in RNA secondary structures such as Ribozymes [14]. During 1980s *Pseudoknots*

were described to be a section of structures in virus RNAs in plants, which resemble tRNAs. Several important biological processes rely on RNA molecules with pseudoknots. For example, the RNA component of human telomerase contains a pseudoknot that is critical for activity [16]. The structures formed by pseudoknots are also crucial building block of RNA tertiary structure. Hence it is of importance to predict not only simple motifs but also *Pseudoknots*.

Pseudoknots can be defined as follows: If R is an RNA sequence such that $R = r_1, r_2, r_3, r_4 \dots r_n$, and $(r_x r_y)$ and $(r_p r_q)$ are two different base pairs existing in this RNA structure ($x < y$ and $p < q$), a pseudoknot is composed of these two base pairs when $1 \leq x < p < y < q \leq n$ [13]. Simply stated, a *Pseudoknot* is a nucleic acid secondary structure containing at least two stem-loop structures in which a section of one stem is intercalated between the two sections of another stem. Pseudoknots fold into knot-shaped three-dimensional conformations but are not true topological knots.

3 MARSs

In this section, we will describe our MARSs algorithm from a design perspective and contrast it with Dynamic Programming algorithms used for RNA secondary structure prediction.

3.1 Overview

MARSs uses a top-down prediction methodology unlike Dynamic Programming algorithms that employ a bottom-up approach. In a top-down approach motifs are generated on a global scale and then the local regions are explored. MARSs stores all the intermediate results, as with Dynamic Programming algorithms, in order to prevent duplication of effort. Unlike Dynamic Programming based algorithms MARSs does not view the secondary structure prediction as a set of overlapping sub-problems. This key distinction makes MARSs high-scalable and easily portable to architectures that have large number of parallel processing units.

Given a primary RNA structure, MARSs produces several possible RNA secondary structures. This is also a key difference between MARSs and Dynamic Programming based algorithms where the latter typically produces a single secondary structure. MARSs has several tuning parameters to its base-pair maximization engine that can be adjusted to produce varying number of output structures. A bioinformatician may use these to inject knowledge from wet-lab experiments and impact the behavior of MARSs algorithm. We believe this attribute makes MARSs algorithm future-proof.

MARSs algorithm does not employ a dictionary-based approach in identifying local structural motifs. Yet, it can predict all known RNA structural motifs and by extension is capable of finding newer types of motifs - either composite of known motifs or entirely newer ones.

MARSs is engineered to be parallelized and consists of several independent processes. Therefore, unlike Dynamic Programming based algorithms, MARSs

can be easily parallelized and significant speed up can be achieved when executed on high performance computers with multiple cores. MARSs consistently shows a speed up of close to N , where N is the number of cores.

MARSs also shows high levels of prediction accuracies. In our experiments comprising of 100 real sequences with verified 2° structures, MARSs produced an average PPV (Positive Predicted Value) of **76.46%** and the average sensitivity was **81.04%**. Hence, MARSs is ready to be used with RNA sequences with unknown 2° structures. These PPV and Sensitivity values compares well with other State-of-the-art algorithms (see results section).

3.2 Matrices and Folding

The foundation of the MARSs algorithm is basically built upon two matrices: namely the **Base Pair matrix (or BP matrix)** and the **Affinity Matrix (or AM)**. The BP matrix remains static throughout the running of the algorithm. It basically represents the base-pairing affinities between all possible nucleotides. It is a 4x4 matrix with the row and column representing known RNA alphabets. The base pair matrix used in our experiments is shown in Figure 1.

A-A 0	A-C 1	A-G 0	A-U 2
C-A 1	C-C 0	C-G 2	C-U 0
G-A 0	G-C 2	G-G 0	G-U 1
U-A 2	U-C 0	U-G 1	U-U 0

Fig. 1. Base Pair Matrix

Fig. 2. Folding across 10 and 14 and attempted bonds

The score of each of the possible base pairs in the BP matrix are assigned using general base pairing rules like *Watson-Crick*, *Hogsteen*, *Wobble* etc. In the above case, *Watson-Crick* pairs (**G-C** and **A-U**) are given a score of 2 while *Hogsteen* (**A-C**) and *Wobble* (**G-U**) base pairs are given a score of 1. A base pair in the BP matrix has a score of 0 when the base-pairing is not possible. The table can be updated with precise values of bond probabilities between the base pairs and a re-run of MARSs will predict the newer structures.

The given RNA primary structure, i.e. the nucleotide sequence, and this static BP matrix is what is used to initially construct the AM. This AM is initialised during runtime. For a given sequence of nucleotides, each nucleotide, from one end to the other, are numbered from 0 to $N - 1$, where N is the length of the nucleotide sequence. This numbering is like array notation in computer programming languages so we can refer to any specific nucleotide in the sequence using integers 0 to $N - 1$. Now we proceed to construct an $N \times N$ matrix (which is the AM), the row and column numbers of which refer to the nucleotide numbers as

mentioned. Each element at co-ordinates (x, y) of the AM represents the score of a possible base pairing between nucleotide number x and nucleotide number y . This matrix is filled by referring to the BP matrix.

In our experiments, we used real pseudoknot structures retrieved from the pseudoknot database called *Pseudobase*. Let us take, for example, a pseudoknot present in the RNA of brome mosaic virus, having a PKB-number of **PKB155** in *Pseudobase* [14].

The AM consists of all possible interactions between nucleotides in the RNA sequence. Hence we hypothesize that all the possible secondary structures are present in this master matrix. By crawling this matrix in specific pre-determined ways we can extract the correct structures of this sequence at much less computational cost than dynamic programming. The RNA consists of nucleotides which are strung together naturally in a chain. Hence base-pairing is not possible between two consecutive nucleotides in the sequence. Thus the next step in the algorithm is to zero all the consecutive nucleotides' interactions in the AM, which we call the Neighbor pair rule. Mathematically, they are elements of co-ordinates $(i, i + 1)$ and $(i - 1, i)$, where $0 \leq i \leq N - 1$.

As the RNA secondary structure is formed by the primary sequence folding upon itself, the next step we do in our algorithm, is to fold the RNA sequence in various ways. This initial folding can give rise to hair pin loops. The folding can occur across any pair of nucleotides in the RNA sequence. Since all nucleotides can pair with each other, the number of possible folds are: $N(N - 1)/2$. In the AM, each element can also be considered to be a folding point. For example, figure 2 shows the shape of the RNA after we just fold it across bases number 10 and 14, including all the attempted bonds (explained later).

3.3 Base Pairing - Symmetric Fold

The next step is to form base pairs in the exposed local regions using the first fold as the anchor. We have added two algorithms for this purpose. The algorithms produce secondary structure with symmetrical and asymmetrical structural motifs. In this way, we are able to generate structures that consist of all types of known structural motifs.

The first of the two base-pairing algorithm is called **S-fold (or Symmetric Fold)** base pairing. For each of the folds, we try and form bonds along bases that are directly facing each other after the fold. For example, in Figure 2, the bases along which the fold has been made are numbered (in this case 10 and 14), and the *orange* lines are the base-pairings that our S-fold base pairing algorithm will try to form. The bases at both ends of the primary sequence that never form bonds with any other base, contribute to form potential dangling ends. In this case these bases are the ones with no orange line attached to it. When no bonds can be formed between opposing nucleotides and are surrounded by base pairs, internal loops are born. To form a base pair, we refer to the AM element corresponding to those two nucleotides, and if it is a value other than 0, we add the score to the score of that structure and form a base pair. Once a base pair is formed, we also assign 0 to the entire corresponding rows and columns in the AM of the nucleotides involved in

the base pair. This is done as these two nucleotides cannot bond with any other nucleotides in this particular fold. This process effectively means we are traversing the AM along a 45 degree line, starting from the folding point and forming bonds whenever the element has a value of greater than 0. An AM numerical representation of the folding starting from nucleotides 10 and 14 is shown in Figure 3. The *dark green* shaded element is the folding point, and the *blue* elements are the base pairs that are formed in case of S-fold. The *grey* area is the hair-pin loop. The *light brown* elements are represents the dangling ends. The *red*, *orange* and *yellow* colored elements represent strong, weak & no bonds and are not used in this level 1 secondary structure.

Fig. 3. AM representation of folding and S-fold bonding across bases 10 and 14

Fig. 4. Base Pairs tried in A-fold

Using this method, multiple anchor folds are formed and for every anchor fold the possible base-pairs are formed. At the end of the process, we have a set of structures that we refer to as level 1 secondary structures. These structures do not have pseudoknots yet, but have hair-pin loops, stems, internal loops and possible dangling ends.

3.4 Base Pairing - Asymmetric Fold

The second scheme, Base Pairing - Asymmetric Fold scheme, can predict all types of level 1 structural motifs that have some sort of asymmetry about them, like bulges and asymmetric internal loops. Once the anchor fold is determined, this base pairing algorithm using two pointers begins traversal along the two *arms* about the fold. As an example, suppose a fold is determined by the folding algorithm to be at nucleotides (say 10 and 14), then this A-fold base pairing would place two pointers at the two folding nucleotides, as shown in Figure 4.

So in this example, let us assume that the fold is made across the *orange* and *blue* circled bases. The circles represent the pointers, and hence the two pointers are based at these two nucleotides. Now these two pointers try forming base pairs in the following manner -

Fig. 5. Level 1 structure after A-fold base pairing across 10 and 14 of PKB155

Fig. 6. AM representation of Level 1 A-fold structure

1. The pointer tries to make a base pair with the base of the opposite pointer
2. If the AM does not support such a base pair, it moves one base along the opposite arm and tries again.
3. If AM still does not support such a base pair, it again moves one base, and keeps moving till a pre-defined limit is reached. In our case we placed a limit of 6 nucleotides, since we observed a maximum bulge of 6 nucleotides in *Pseudobase* [14]. If limit is reached, then it does not form any base pair.
4. Both the pointers follow the above steps. Figure 4 shows the base pairs that are attempted initially by the pointers in our hypothetical example.

The blue pointer tried only 2 base pairs, since the second one forms a base pair (A-C, *Watson-Crick*) and hence the pointer stopped traversing. While the orange pointer had to traverse to the limit (6 nucleotides) until it could find a possible base pair (A-C, *Watson-Crick*). So, following the above procedure, each of the two pointers will produce none or one base pair. Therefore, we need a protocol or precedence to decide which base pair to choose if both pointers each form one base pair? We only need to make a choice in the scenario (common) that both pointers yield a different base pair. We follow the criteria below in order of preference in order to decide the final base pair.

1. Choose the base pair that had to skip least number of nucleotides to form the base pair. We call this *distance*.
2. If *distance* is same, then we pick the base pair which has more weight, i.e. a *Watson-Crick* base pair is chosen over a *Hogsteen* base pair.
3. If both the base pairs' weight are the same, we arbitrarily choose one pointer.

If a base pair is formed, the two pointers each then move to the nucleotide next to the base-pair just formed and if no base pair is formed the pointers move one base away from its previous position. Then the whole process is repeated until one of the pointers reach the end of the strand. At the end of this process, a level 1 structure is produced by A-folding. Figure 5 shows the level 1 structure produced by A-folding for our hypothetical sequence with the anchor fold at nucleotides

across 10 and 14. We can now observe that a bulge is formed, hence asymmetry is introduced in this algorithm. Figure 6 shows the AM representation of this structure. We can see that the traversal is no longer 45 degrees from the folding point, but made of broken lines, to introduce asymmetry.

3.5 Level 2 Folding

Now the next step is to form Level 2 secondary structural elements, which is Level 2 folding. We currently have generated N^2 different level 1 structures corresponding to folds. As mentioned earlier, for each of these structures we change the AM by marking the rows and columns corresponding to the nucleotides involved in base-pairs in the Level 1 structure to be 0, since they can no longer interact with other nucleotides. So now, in order to form Level 2 secondary structural elements, we *fold* the Level 1 structures again and run S-fold base-pairing for each level 1 fold to find more base pairs that can cause pseudoknots or coaxial stacks. A level 2 folding may or may not result in a pseudoknot, as shown in Figure 7.

Fig. 7. Illustration of pseudoknots or coaxial stacks

Fig. 8. One predicted structure of PKB155

Let us go back to our test case of Level 1 structure with folding across 10 and 14. After Level 2 folding, one of the high energy (16) (and therefore more likely secondary structure) is as shown in Figure 8. This figure shows Level 2 (level 2 folding across 4 and 9) structures and the corresponding Level 1 [10, 14] structure. All bonds are correctly predicted when compared to the structure in Pseudobase [14], hence the predicted structure has a base-pair distance of 0. It has a PPV of 100% and a sensitivity of 100%.

4 MARSs Complexity Analysis

In this section, we attempt to derive the time and space complexities of MARSs algorithm. It should however be noted that the actual space and time complexities depends on the composition of the nucleotides themselves that in turn

affects the set of possible structures. The resource complexities also depends on the tuning parameters that dictate the final number of secondary structures that are desired.

$$\text{Number of folds} = \frac{N(N-1)}{2}$$

Hence, Space complexity is $O(n^2)$

$$\text{Maximum number of potential base pairs traversed in S-Fold} = \frac{N}{2}$$

$$\text{Minimum number of potential base pairs traversed in S-fold} = 1$$

$$\text{Average potential base pairs traversed} = \frac{N+2}{4}$$

$$\text{TIME COMPLEXITY} = \frac{N(N-1)}{2} \times \frac{N+2}{4} = O(n^3)$$

Since A-fold base pairing simply increases the number of base pairs traversed by a constant, hence complexity remains same.

5 Results

5.1 Accuracy Measures

As mentioned before, the accuracy measures that we use to analyze the accuracy of a predicted structure when compared to experimentally verified structures are **Sensitivity** and **PPV**. **Positive Predictive Value (PPV)** is the number of correctly predicted base pairs as a percentage of the total number of base pairs in the predicted structure. Its primary focus is on the accuracy of predicted base pairs, without regard to any unpredicted base pairs. **Sensitivity** is the number of correctly predicted base pairs as a percentage of the total number of base pairs in the experimentally verified structure. Its primary focus is on predicting base pairs present in the actual structure, without regards to the number of false base pair predictions. These two measures are now the standards for measuring accuracy in case of RNA secondary structure prediction [15].

$$\text{PPV} = \frac{\text{number of correctly predicted base pairs}}{\text{total number of base pairs in PREDICTED STRUCTURE}} \times 100\%$$

$$\text{Sensitivity} = \frac{\text{number of correctly predicted base pairs}}{\text{total number of base pairs in ACTUAL STRUCTURE}} \times 100\%$$

A structure can be said to be perfectly predicted, when both the PPV and sensitivity values are 100%. PPV and sensitivity shows the measure of accurate base pairs predictions relative to the predicted and the actual structure respectively.

5.2 Accuracy of MARSSs

For our analysis, we selected a set of 100 sequences from *Pseudobase* [14] from different classes as classified in the website. The average PPV of all these sequences is **76.46%** and the average sensitivity is **81.04%** and the average normalized b-p distance is **12.58%**.

It can be seen that the average PPV and sensitivity of the MARSSs algorithm on the randomly selected sequences is pretty good. In order to be comprehensive, we also compare the predictions with a dynamic programming based RNA secondary structure algorithm and observe the results. We ran the same sequences on *Ruan's server* (which predicts Pseudoknots) [9] that employs the Ruan's algorithm as previously mentioned and also in the software *PKNOTSRG* (as previously described) [7]. Ruan's algorithm showed an average PPV and sensitivity of **56.8%** and **58.9%** respectively, while PKNOTSRG had an average PPV and sensitivity of **70.5%** and **67.8%** respectively. Hence we can see MARSSs produced significantly better results.

Fig. 9. Ruan et al. and PKNOTSRG compared with MARSSs

Figure 9 shows a bar chart with Ruan's and PKNOTSRG's accuracy parameters compared with MARSSs, for RNA of different classes as classified by *Pseudobase* [14]. We can see that MARSSs appears to be consistently more accurate than Ruan's algorithm and PKNOTSRG in terms of all classes of RNA, except for a few instances like Aptamers, Viral 3' UTR and Viral Ribosomal Frame-shifting signals. Even in those cases it is only slightly less accurate than the other algorithms only and the difference is a very small amount.

6 Parallelization of MARSSs

One of the primary design objectives of MARSSs is to be easily parallelized and take advantage of the explicit hardware parallelism available on the hosts. This is achieved as follows. After the first Level 1 folding, we obtain a series of Level

Fig. 10. Multi-core implementation of MARSS in Intel Quad Core Xeon 2992.006 MHz

Fig. 11. Colour Mapped Surface for execution times

1 structures, each having its own Affinity Matrices. Hence, when we do Level 2 folding of these structures, they are completely independent of each other. Therefore, we can run each of the Level 2 folding activity on different cores concurrently. The data plot in figure 10 demonstrates how the algorithm speeds up when we run it on multiple cores, while processing different sequence lengths. For the plot, the algorithm was run in Intel Quad Core Xeon machine with a clock speed of 2992.006 MHz.

From figure 10, we can clearly see that as the number of cores increased, the execution time significantly decreases. The entire set of 100 sequences from *Pseudobase* [14] was then run on a second machine. This machine is a virtual machine having 16 virtual cores being run by the software QEMU version 0.9.1. This virtual machine is actually powered by 12 physical cores, each being Intel Xeon CPU E7450 2.40 GHz. The graph in figure 11 shows a colour mapped surface plot of execution time vs number of cores vs sequence length. We assigned more colour codes for lower times, since the trend is less apparent for the subtle change in shape of the graph. We can see the clear trend in decreasing execution time as number of cores are increased due to parallelization, and also we can see for smaller sequences, too high number of cores tend to slow it down. Also, for large sequence lengths, we can see the enormous significance of decrease in run time, as number of cores are increased.

7 Conclusion

In this paper, we have introduced a new high-performance and scalable algorithm for RNA secondary structure prediction. The algorithm is designed to be parallelized and strives to be fundamentally different when compared to the generic class of RNA secondary structure prediction algorithms that are Dynamic Programming based, thus pushing the state-of-the-art to the next level. The algorithm is shown to take complete advantage of the explicit hardware

parallelism available with our experiments. We have conducted experiments to quantify the performance of the algorithm. For this, we extracted RNA sequences with experimentally verified structures from Pseudobase and compared MARSS predictions with actual structures. We have also conducted a 2nd experiment with a focus on performance improvement through parallelization. For this, we employed a 16 core Intel 64-bit server. The results indicate that the algorithm produces much better results when compared with other algorithms and also is scalable to use hardware parallelization features. The algorithm however is still in its early stages and extensive experiments are to be conducted on actual very large-scale structures to identify its limitations, if any. This would be an immediate extension to the problem under study.

References

1. Nussinov, R., Piecznik, G., Grigg, J.R., Kleitman, D.J.: Algorithms for loop matchings. *SIAM Journal on Applied Mathematics* 35, 68–82 (1978)
2. Waterman, M.S., Smith, T.F.: Rapid dynamic programming methods for RNA secondary structure. *Adv. Appl. Math.* 7, 455–464 (1986)
3. Lyngs, R., Zuker, M., Pedersen, C.: An Improved Algorithm for RNA Secondary Structure Prediction. Tech-report BRICS RS-99-15 (1999)
4. Rivas, E., Eddy, S.: A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.* 285(5), 2053–2068 (1999)
5. Akutsu, T.: Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics* 104, 45–62 (2000)
6. Dirks, R., Pierce, N.A.: A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry* 2003 24, 1664–1677 (2003)
7. Reeder, J., Giegerich, R.: Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics* 5, 104 (2004)
8. Deogun, J., Donts, R., Komina, O., Ma, F.: RNA Secondary Structure Prediction with Simple Pseudoknots. In: APBC 2004, pp. 239–246 (2004)
9. Ruan, J., Stormo, G.D., Zhang, W.: ILM: a web server for predicting RNA secondary structures with pseudoknots. *Nucleic Acids Research* 32(Web Server Issue), W146–W149 (2004)
10. Ren, J., Rastegari, B., Condon, A., Hoos, H.H.: HotKnots: Heuristic prediction of RNA secondary structures including pseudoknots. *RNA* 11, 1494–1504 (2005)
11. Tian, B., Bevilacqua, P.C., Diegelman-Parente, A., Mathews, M.B.: The double-stranded-RNA-binding motif: interference and much more. *Nature Reviews Molecular Cell Biology* 5, 1013–1023 (2004)
12. Brion, P., Westhof, E.: Hierarchy and dynamics of RNA folding. *Annu. Rev. Biophys. Biomol. Struct.* 26, 113–137 (1997)
13. Fu, X.Z., Wang, H., Harrison, W., Harrison, R.: RNA Pseudoknot Prediction Using Term Rewriting. *International Journal of Data Mining and Bioinformatics* (2006)
14. Batenburg, F.H., Gultyaev, A.P., Pleij, C.W.: PseudoBase: structural information on RNA pseudoknots. *Nucleic Acids Research* 29(1), 194–195 (2001)
15. Ding, Y., Chan, C.Y., Lawrence, C.E.: RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. *RNA* 11, 1157–1166 (2005)
16. Chen, J.L., Greider, C.W.: Functional analysis of the pseudoknot structure in human telomerase RNA. *Proc. Natl. Acad. Sci. USA* 102(23), 8080–8085 (2005)