compPknots: a Framework for Parallel Prediction and Comparison of RNA Secondary Structures with Pseudoknots

Trilce Estrada[1], Abel Licon[1], and Michela Taufer[1]

[1]Computer Science Department, University of Texas At El Paso,
El Paso, TX 79968
{tpestrada, alicon2, mtaufer} @ utep.edu

Abstract. Codes for RNA secondary structure prediction based on energy minimization are usually time and resource intensive. For this reason several codes have been simplified: in some cases they do not predict complex structures like pseudoknots, other times they predict structures with reduced lengths, or with simple pseudoknots. Each of these codes has its strengths and weaknesses. Providing scientists with tools that combine the strengths of the several codes is a worthwhile objective. To address this need, we present compPknots, a parallel framework that uses a combination of existing codes like Pknots-RE and Pknots-RG, to predict RNA secondary structures concurrently and automatically compare them with reference structures from databases or literature. In this paper compPknots is used to compare the predictions of 217 RNA structures from the PseudoBase database. Its parallel master-slave architecture provide scientists with higher prediction accuracies in shorter time.

1 Introduction

Nucleic Acid chains called RiboNucleic Acids (RNA) play critical roles in several processes in living organisms. In cellular protein synthesis, genetic information is expressed through RNA chains. In some viruses, RNA chains are carriers of genetic codes. RNA molecules are composed of 4 types of nucleotides or bases: adenine (A), cytosine (C), guanine (G) and uracil (U) that fold back on themselves thus pairing with each other. So for example C-G and A-U form stable *base pairs* with each other through the creation of hydrogen bonds between donor and acceptor sites on the bases. The *secondary structure* of an RNA molecule is the collection of base pairs. Since the experimental identification of RNA secondary structures is time demanding, in the past decades a significant effort has been made to build RNA structure predictions from sequence data using computational methods. A first approach consists of the computation of common foldings for a family of aligned, homologous RNAs. Usually, the alignment and secondary structure inference must be performed simultaneously, or at least iteratively and therefore methods that employ this approach for their predictions are not easy to automate and require significant human intervention. A second approach targets the structure prediction of single sequences based on the minimization of the free energy of a folding [1]. With the significantly increasing computing power, today's methods that employ this approach can be easily automated and therefore are attractive methods to perform these predictions. Several motifs can be commonly found in RNA secondary structures: stem-loops (i.e., helix, hairpin loop, interior loop, buldge loop, multi loop) and pseudoknots.

Among the several motifs in an RNA molecule, the prediction of pseudoknots is particularly demanding in terms of data and computational power required for codes based on energy minimization methods. Therefore several codes that use the minimization of the RNA free energy of a folding often do not include pseudoknot predictions [2]. Because pseudoknots have been observed in several RNA molecules [3] omitting them from predictions can significantly affect the prediction accuracy. To reduce computation time and data storage, several codes that include pseudoknots have been implemented with significant simplifications: the Rivas and Eddy code (Pknots-RE) [4] can predict the secondary structure of very short RNA segments (of the order of hundreds of nucleotides) while RNA molecules are normally compounds of thousands of nucleotides; Reeder and Giegerich [5] have significantly reduced the complexity of the pseudoknots that their code, Pknots-RG, can predict. To solve the limitations of these codes on their own, a more effective approach would be to use the combination of both. Providing the user with a tool that combines the strengths of each of the single codes would be a worthwhile goal. Once predictions are performed, a comparison of the predicted secondary structures against experimentally observed structures is needed. The comparison of predicted RNA secondary structures is often performed manually using tools such as PseudoViewer [6]. This manual task is monotonous, highly sensitive to errors, and when the sequences are too long or too numerous, it is impossible for a human to do this in a feasible amount of time.

The above listed critical aspects point out the need for tools that (1) predict secondary structures of RNA segments, included pseudoknots, in parallel using combinations of energy based methods and (2) automatically compare the predictions against reference structures from databases or literature. We address this need in this paper by presenting a parallel framework, *compPknots*, for prediction and comparison of RNA secondary structures with pseudoknots. *compPknots* exploits the advantages of parallel computation using the MPI library MPICH to predict large numbers of RNA secondary structures using well-known RNA structure prediction codes such as Pknots-RE [4] and Pknots-RG [5] concurrently. Using *compPknots*, we evaluate the prediction accuracy of the single Pknots-RE and Pknots-RG predictions against their combined accuracy for a set of 217 RNA segments from the PseudoBase database [7]. Running the predictions in parallel on a Beowulf cluster using these two codes significantly reduced the execution time for the predictions of the 217 RNA segments.

The rest of the paper is organized as follows: In Section 2, we present an overview of the main biochemical concepts needed to understand this work and a short review of works in the field of RNA structure prediction. Section 3 describes the *compPknots* framework, its software components for parallel prediction and comparison of RNA secondary structures, and how to use it. Section 4 presents the evaluation of *compPknots* in terms of its prediction accuracy and performance. In Section 5 we conclude and present current work in progress.

2 Background and Related Work

A ribonucleic acid (RNA) is one of the two types of nucleic acids (Deoxyribonucleic acid DNA and Ribonucleic acid RNA) found in living organisms. An RNA molecule represents a long chain of monomers called nucleotides or bases. RNA contains four different nucleotides: adenine, guanine, cytosine, and uracil that are represented with the letters A, G, C, and U respectively. A sequence of these bases is strung together to form a long, single-stranded RNA molecule. The molecule, whose sequence may be up to thousands of bases long, tends to fold back on itself, mostly by pairing between complementary bases: C and G form a complementary base pair, and so do A and U. The *secondary structure* of an RNA molecule is the collection of base pairs that occur in its three dimensional structure. RNA secondary structures can be classified into two basic categories called *stem-loops* and *pseudoknots* (see Fig. 1). Both kinds of secondary structures on overlapping RNA viral genes have been implicated in important viral gene expression processes [8]. Pseudoknots have been also shown to be relevant in many RNA mediated processes. Examples are the self-splicing group I introns [9], ribosomal RNAs, or RNaseP. Recently, pseudoknots were located in prion proteins of humans, and confirmed for many other species [10].

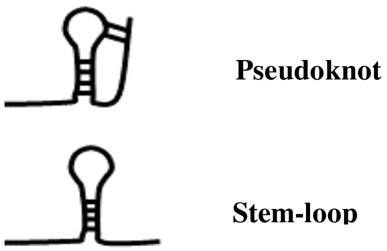


Fig. 1. Pseudoknot and stem-loop

With the current increased interest in the RNA functions, algorithmic support for analyzing structures that include pseudoknots is much in demand; determining such structures, including pseudoknots, has been shown to be an NP-hard problem. Only for more restricted classes of pseudoknots, polynomial algorithms have been implemented. Rivas and Eddy [4] developed a dynamic programming algorithm, Pkno ts-RE, for predicting optimal (minimum energy) RNA secondary structures, including pseudoknots. The algorithm has the worst case time and space complexities of $O(n^6)$ and $O(n^4)$ respectively. The implementation of the algorithm uses standard RNA folding thermodynamic parameters augmented by a few parameters describing the thermodynamic stability of pseudoknots and by coaxial stacking energies. Reeder and Giegerich [5] improved the complexity of the algorithm, reaching the $O(n^4)$ space and $O(n^2)$ time, by using the Minimal Free Energy (MFE) model. The runtime improvement, compared to Pknots-RE, results from an idea of canonization, while the space improvement results from disallowing chained pseudoknots. Uemura et al. [11] proposed an algorithm based on tree-adjoining grammar. The time complexities of their algorithm depends on the types of pseudoknots: it is $O(n^4)$ for simple pseudoknots and O(n^5) or more for the other pseudoknots. Although the algorithm can always find optimal structures, tree-adjoining grammars are complicated and impractical for longer RNA sequences. Akutsu [12] analyzed Uemura's method and found that the tree-adjoining grammar was not crucial but the parsing procedure was. Since the parsing procedure is intrinsically a dynamic programming procedure, Akutsu reformulated this method as a dynamic programming procedure without the tree-adjoining grammar. This method has not been implemented into a code yet.

compPknots aims to include a variety of codes to capture the strength of each single code. The current version of our framework presented in this paper includes Pknots-RE and Pknots-RG, but it can be easily extended to accommodate other existing codes for RNA predictions.

3. Components, Parallelization, and Usage of *compPknots*

compPknots is a framework that can integrate concurrent executions of existing codes for RNA secondary structure predictions such as Pknots-RE and Pknots-RG, with the capability of automatically measuring the level of prediction accuracy for these codes by using a comparison approach based on stacks. The framework is written in C and employs the MPICH library for concurrent predictions and comparisons. Its modular structure allows users to easily extend it to accommodate other codes and comparison techniques.

The current framework integrates predictions using Pknots-RE and Pknots-RG and their comparisons. For each input segment, the chain of nucleotides, or RNA segment, is read and its correctness is checked: this checking consists of verifying that only characters in the following list compose the chain: A, C, G, U, a, c g, and u. If comparisons are scheduled, the user provides *compPknots* with the file with the experimentally observed structures and *compPknots* schedules the corresponding comparisons. The observed secondary structures are traditionally represented in terms of strings of brackets, i.e., "(" and ")","[" and "]", "{" and "}", dots ".", and colons ":". Two paired nucleotides are represented with two closed brackets collocated in the string at the same position as the correspondent nucleotides in the input segment. A checking is required for the observed secondary structure to control whether it contains only characters in the following list: "[", "]", "{", "}", "(", ")", ".", and ":". Also predicted secondary structures are returned as a string of braces, parenthesis and dots. After checking the correctness of input segments and observed structures, *compPknots* can execute either Pknots-RE or Pknots-RG or both (default configuration). Finally, predicted secondary structures are compared with the observed structures provided in the input file and the statistics are then printed to the screen or stored to a file.

The comparison between a predicted structure and an observed structure is based on stacks: for the predicted structure and the observed structure the code allocates a pair of stacks for storing general stem-loops (predicted stem-loop stack and observed stem-loop stack) and a second pair for storing loops associated to pseudoknots (predicted pseudoknot stack and observed pseudoknot stack). Each pair of stacks is used twice for each assignment in case a pseudoknot is present. If a pseudoknot is not present, only the pair of stacks associated to the stem-loop is used. One bracket representing one of two paired nucleotides, e.g., “(“ and “)” or “[“ and “]”, or one dot or colon, i.e., “.” or “:”, representing an unpaired nucleotide, is read at the time from both the predicted and observed structures. If an opening character appears, e.g., “(“ or “[“, its position in the RNA segment is stored in the corresponding stack based on the fact that the character is from the predicted or from the observed structure as well as on the fact that the code is going through a first set of parenthesis (a stem-loop) or is going through a second set without having completed a previous one (a pseudoknot). If the character is a closing one, e.g., “)“ or “]“, then the last element in the corresponding stack is removed. If a removal occurs at the same time from the predicted and observed stacks, the equivalent nucleotide positions are compared. If the positions are equal, the number of true predicted base pairs is incremented by one; otherwise the system increments the number of false predicted base pairs.

compPknots uses a master-slave paradigm to run predictions and comparisons, where the master is in charge of getting new jobs and dispatching them to available slaves. The master validates the correctness of RNA segments submitted for prediction as well as the correctness of observed RNA secondary structures before submitting them to the slaves. The master also receives the results of the predictions and comparisons and prints or stores them. Slaves are activated by the master that sends them the initialization parameters, e.g., what code to use for the prediction and whether to compare predictions with observed structures. Once the slave is active, it starts its prediction cycle in which it requests and gets new jobs. The cycle terminates if there is no job remaining. Fig. 2 shows the flow of tasks executed by the master and the slaves. The assignment of a single job at a time to the hosts helps prevent load imbalances: the length of jobs depends on the length of their RNA segment and whether the resources on the distributed system are dedicated or not.

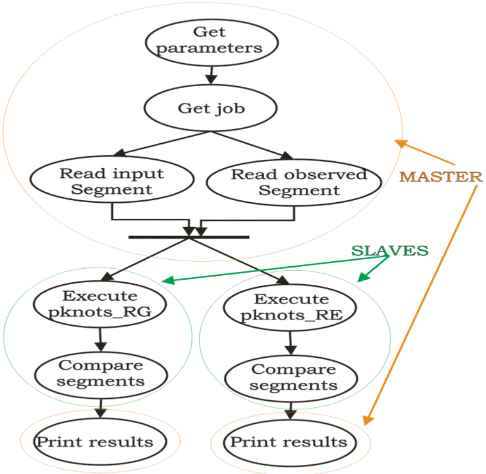


Fig. 2. Concurrent prediction and comparison in *compPknots*

compPknots is flexible in terms of capabilities that scientists can select for a given run. The user can, for example, choose whether to predict and compare a nucleotide segment provided to the code from the command line or several segments listed in a file, or whether to select just one of the RNA structure prediction codes and store the output to a file rather than printing it to the screen. A list of capabilities and possible input options are reported in [13]. For each RNA segment, the following metrics are returned to the scientist:

- Length of the RNA segment: Number of nucleotides or bases in the given segment.
- Total observed base pairs (*total pairs*): Number of base pairs in the observed secondary structure.
- Total predicted base pairs: Number of base pairs in the predicted secondary structure. This is the sum of true pairs and false pairs.
- True predicted base pairs (*true pairs*): Number of pairs predicted that are also in the observed secondary structure.
- False predicted base pairs (*false pairs*): Number of base pairs predicted that are not in the observed secondary structures. This is the sum of wrong predicted pairs and true pairs not detected.
- Sensitivity: Number of true pairs over the number of base pairs.

$$sensitivity = \frac{true\_pairs}{total\_pairs}$$

- Selectivity. Number of true pairs over the sum of true and false pairs.

$$selectivity = \frac{true\_pairs}{true\_pairs + false\_pairs}$$

- Total Energy: Energy of the RNA secondary structure.

The sensitivity and selectivity are two standard metrics that are used by scientists to quantify the accuracy of a prediction [14]. The sensitivity indicates whether the prediction has captured the secondary structure partially or completely. The selectivity indicates whether a prediction has introduced additional base pairs in the predicted secondary structure that are not present in the observed secondary structure. These metrics range from 0, i.e., the structure has been completed miss-predicted, to 1, i.e., in the case of a successful prediction. More in particular, if the sensitivity is 1, it means that all the observed base pairs have been correctly predicted. If the selectivity is 1, then no additional base pair has been predicted.

4. Evaluation

The evaluation of *compPknots* in this paper consists of two components: first of all, the framework is used to evaluate the effectiveness on the prediction accuracy of combining several prediction codes; then the performance analysis of predictions using each single code sequentially and the combination of both concurrently is evaluated.

For the *evaluation of the accuracy* of single codes, Pknots-RE and Pknots-RG, and their combination, we used 217 RNA segments and their experimentally observed secondary structures from the PsudoBase database choosing all the complete segments in this database, i.e., those segments that do not present nucleotides gaps in their brackets representation [7]. For the predictions with Pknots-RG, we scored the predicted structures exclusively on an energy base and we did not force pseudoknot identifications as possible in this code [5]. For the evaluation of the prediction accuracy of the single codes (considered separately), we considered 4 levels of sensitivity and selectivity, i.e., 0.0, $\lceil 0.0 \rceil - \lceil 0.5 \rceil$ in which the values 0.0 and 0.5 are not included in this range, $\lfloor 0.5 \rfloor - \lceil 1.0 \rceil$ in which the value 1.0 is not included in this range, and the last level being 1.0. We counted the number of predictions that fell into each level for the two metrics. Fig. 3 shows the number of predictions for the 4 levels for both Pknot-RE and Pknot-RG. As we can see in Fig. 3, sensitivity and selectivity in Pknots-RE and Pknots-RG have similar behaviors. With reference to the sensitivity, both codes have a significant number of structures that are completely miss-predicted i.e., the codes are not able to capture any base pairs. More in particular, for Pknots-RE, 6.4% of the predictions have no true pairs and 33.6% of the predictions capture all the true pairs. For Pknots-RG, 4.6% of the predictions are completely miss-predicted (which is less than for Pknots-RE) but fewer predictions capture all the true pairs (29.9% for Pknots-RG versus 33.6% for Pknots-RE). The remaining structures are partially predicted correctly, i.e., 59.9% for Pknots-RE and 65.4% for Pknots-RG. With reference to the selectivity, both the codes show the tendency to predict more base pairs than those observed experimentally: only for 29 RNA segments predicted by Pknots-RE (13.3% of the predicted secondary structures) the number of false pairs is zero. For RNA segments predicted by Pknots-RG only 34 predictions (15.6% of the predicted secondary structures) have no false pairs. In general with reference to the overall pool of RNA segments, we found that Pknots-RE has an average sensitivity of 73.3% and an average selectivity of 62.2% while Pknots-RG performs slightly better with an average sensitivity of 75.6% and an average selectivity of 64.7%.

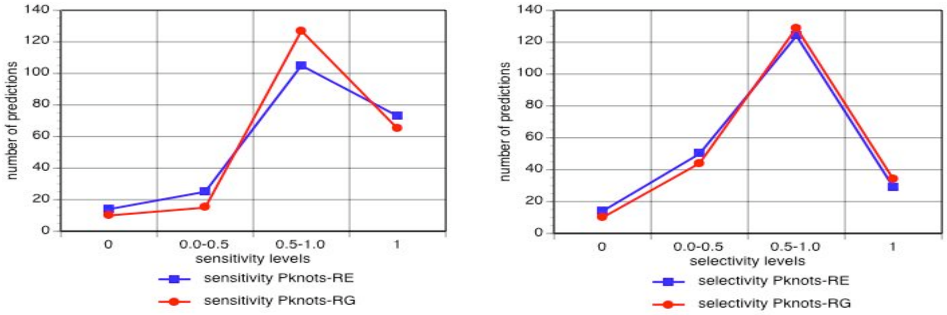


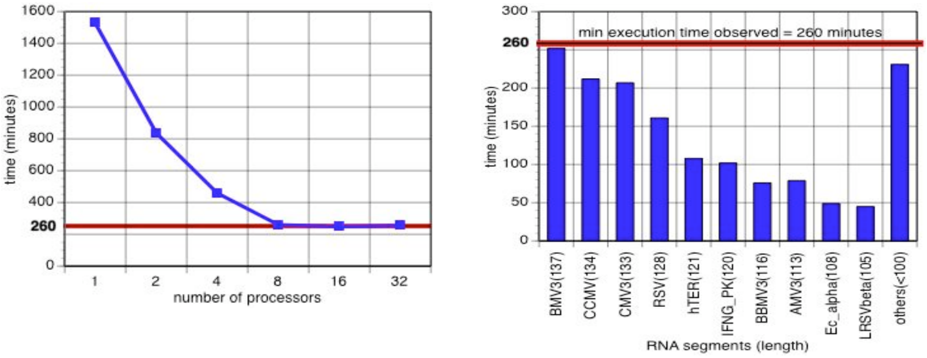
Fig. 3. Sensitivity and selectivity levels for 217 predicted RNA secondary structures using either Pknots-RE or Pknots-RG for predictions.

If we consider the single predictions separately we can see that in several cases one of the two codes provides an accurate prediction while the other code performs poorly. An evaluation of the accuracy using a combination of both codes was performed using *compPknots* and the results are reported in [13]. In this comparison we combined different levels of sensitivity and selectivity of secondary structures predicted using the two codes, Pknots-RE and Pknots-RG concurrently. For each code we considered the follow levels: (1) both sensitivity and selectivity are zero, i.e., 0{0}; (2) sensitivity and selectivity range within zero and one, i.e., 0.x{0.y}; (3) selectivity is one but selectivity is less than one because, despite the prediction capture all the true pairs, it also captures additional pairs that have not been observed experimentally (false pairs), i.e., 1{0.y}; and selectivity and sensitivity are both one, i.e., 1{1}. We observed that the cross prediction using both the codes increased the prediction accuracy significantly: we were able to predict correctly the complete secondary structure of 92 RNA segments for which the sensitivity was equal to one. This is equal to 42.3% of the pool of RNA segments considered. Also the selectivity was significantly improved: 22.2% of the secondary structures have both sensitivity and selectivity equal to one. The average sensitivity and selectivity were increased to 82.4% and 70.8% respectively proving how the combination of prediction codes can indeed increase the final accuracy for our 217 RNA segments

Another important aspect of running the codes concurrently is the *execution time*. To address this issue, we ran the predictions using *compPknots* with each code as well as their combination on a Beowulf cluster at the University of Texas at El Paso. The cluster has a head node with 2 AMD Opteron processors, 4 GB memory, 3 TB of disk space (shared by all the nodes over NFS) and 64 compute nodes with 2 AMD Opteron processors, and 4 GB of memory each. Each prediction of the 217 RNA segments was repeated three times and the values reported are average times. The same set of 217 RNA segments were computed using *compPknots* with 1, 2, 4, 8, 16, and 32 processors. The segments have a length that ranges from 21 to 137 nucleotides or bases.

In our performance analysis, we observed that between the two codes, Pknots-RE is the most time and resource intensive. For Pknots-RE, the execution of the sequential prediction and comparison of the 217 RNA segments using one node of the cluster took an average of 1512 minutes, while the same execution using Pknots-RG took only 14.29 seconds. We also observed that the combined execution of the two codes was mainly dictated by two factors: the prediction using Pknots-RE and the execution times of the 10 longest RNA segments. In particular, the latter factor limited the scalability of *compPknots* with this set of RNA segments. Running *compPknots* with Pknots-RE and Pknots-RG with 2 processors took 837 minutes. This time went down to 475 and 260 minutes with 4 and 8 processors respectively. For 16 and more processors the time needed for this set of RNA segments did not scale any further as shown in Fig. 4.a. We measured the times for the prediction and comparison of the ten longest segments with length longer than 100 nucleotides, and compared these times with the total time to predict and compare all 207 remaining segments from our PseudoBase set. Fig. 4.b shows the times to run a Pknots-RE prediction in minutes (x-axis) for different RNA segments (y-axes, where the name of each segment is associated to its length -- number in parentheses). As we can see in the figure, the time for the prediction of segments such as BMV3, CCMV3, and CMV3, whose lengths are 137, 134, and 133 respectively, is comparable with the total time to execute the remaining 207 segments. This suggests that once we have more than 8 processors, those that receive the longest segments will determine the time of the whole simulation. In general, the scalability of *compPknots* executions depends on both the length and number of segments. Even a smart distribution of tasks across processors cannot solve load imbalances due to the executions of predictions in which a few segments dominate over the others because of their much larger length. These observations suggest two future improvement strategies for *compPknots*: the need for the parallelization of the

Pknots-RE code and the potential for *compPknots* to run effectively on clusters of heterogeneous nodes where longer segments are assigned to faster nodes and shorter segments are assigned to slower nodes.



a. Execution time for 217 RNA segments **b.** Segment execution time

Fig. 4. Performance analysis of *compPknots* for a set of 217 RNA segments.

5. Conclusions and Future Work

In this paper we presented *compPknots*, a parallel software framework that given a set of RNA molecules, predicts their RNA secondary structures (including their pseudoknots) using a combination of existing prediction codes concurrently. It also allows users to automatically compare these predictions with reference structures from databases or literature, identifying the predictive accuracy of each RNA secondary structure. *compPknots* allowed us to run two codes concurrently for predictions, Pknots-RE and Pknots-RG, benefiting from the combined prediction capability of both. The combined execution of the codes allowed us to correctly capture a larger number of RNA secondary structures as oppose to using the single codes separately. The parallel framework based on a master-slave paradigm and implemented using MPICH allowed us to achieve results in a shorter amount of time. *compPknots* is a prototype that is part of a "smart" framework that automatically at run time selects RNA segments from large RNA molecule, predicts the secondary structures in parallel using a large variety of existing codes, and based on the comparisons of the collected results selects new RNA segments from the initial RNA chain to ultimately rebuild larger parts of this RNA molecule. We are currently extending *compPknots* to accommodate these features.

Acknowledgments

This material is based in part upon work supported by the Texas Advanced Research Program under Grant No. 003661-0008-2006. Financial support through the National Science Foundation (grant # EIA-0080940, "MII: Graduate Education for Minority Students in Computer Science and Engineering: Extending the Pipeline" and grant # SCI-0506429, “DAPLDS - a Dynamically Adaptive Protein-Ligand Docking System based on multi-scale modeling”) is acknowledged. We wish to thank Dr. Ming-Ying Leung, Prayook Tungjatooronrusamee, and Yash Dahl for their valuable advice during the implementation and testing of *compPknots*.

References

1. Lyngo, R.B. and Pedersen, C.N.S. (2000) RNA Pseudoknot Prediction in Energy-Based Models. *J. of Comp. Biology,* 7(3/4), pp. 409–427.
2. Zuker M. (1989) Computer Prediction of RNA Structure. *Methods Enzymol.* 180, pp. 262-288.
3. Dinman, J.D., Ruiz-Echevarria, M.J., and Peltz, S.W. (1998) Translating old Drugs into new Treatments: Ribosomal Frameshifting as a Target for Antiviral Agents. *Trends Biotechnol.,* 16, pp. 190–196.
4. Rivas, E. and Eddy, S. (1999) A Dynamic Programming Algorithm for RNA Structure Prediction including Pseudoknots. *Journal of Molecular Biology*, 285(5), pp. 2053-2068.
5. Reeder J. and Giegerich R. (2004) Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics*, 5(104).
6. Han, K. and Byun,Y. (2003) PseudoViewer2: Visualization of RNA Pseudoknots of any Type. *Nucleic Acids Res.,* 31, pp. 3432–3440.
7. Batenburg, F.H.D. 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).
8. L. Bidou, G. Stahl, B. Grima, H. Liu, M. Cassan, and J. Rousset: In Vivo HIV-I Frameshifting Efficiency is Directly Related to the Stability of the Stem-loop Stimulatory Signal. *RNA,* 3:1153-1158, 1997
9. Cech, T. (1988) Conserved Sequences and Structures of Group I Introns: Building an Active Site for RNA Catalysis a Review. *Gene*, 73, 259-271.
10. Barette, I., G. Poisson, P. Gendron, and F. Major (2001). Pseudoknots in prion protein mRNAs con_rmed by comparative sequence analysis and pattern searching. Nucleic Acids Research 29 (3), 753-758.
11. Uemura Y, Hasegawa A, Kobayashi S, and Yokomori (1995) Grammatically Modeling and Predicting RNA Secondary Structures. In *Proceedings of the Genome Informatics Workshop*, Universal Academy Press, Tokyo, pp. 67-76.
12. Akutsu, T. (2000) Dynamic Programming Algorithms for RNA Secondary Prediction with Pseudoknots, *Discrete Applied Mathematics,* 104, 45-62.
13. T. Estrada, A. Licon, and M. Taufer: CompPknots: a Framework for Parallel Prediction and Comparison of RNA Secondary Structures with Pseudoknots. *Technical Report UTEP-CS-06-42*, University of Texas, El Paso, September 2006.
14. Gardner P.P. and Giegerich R. (2004) A Comprehensive Comparison of Comparative RNA Structure Prediction Approaches. *BMC Bioinformatics*, 5(140).