Fast Algorithms for RNA Secondary Structure Prediction

A Project Report

Submitted in partial fulfilment of the

requirements for the Degree of

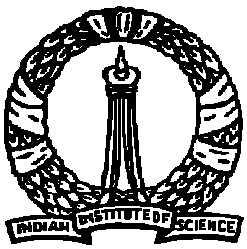
Master of Engineering

in

Computer Science and Engineering

by

Sanjay Vaghela



Computer Science and Automation

Indian Institute of Science

BANGALORE – 560 012

JULY 2006

TO

My Grandparents

Acknowledgements

I thank my guide Dr. Chiranjib Bhattacharyya for his invaluable guidance and support through out my stay in IISc. I am inspired by his dedication to research and his ability to look beyond the scope of a problem domain. I have learned a lot through the several discussions and meetings that we have had. Most of the work done in this thesis has emerged as an outcome of these discussions. I also thank him for his endless patience and trust in me.

I thank the chairman, faculty and the staff of the CSA department.

I thank all my lab-mates: Karthik, Krishnan, Sourangshu and Saketha, who were always helpful for giving useful inputs on any problem that I had faced and also for giving company in lab.

I thank all my friends in IISc: Keyur, Anand, Rajdeep, Debmalya, Rajiv, Akshat, Ramswarup, Mani, PP and all the others who have made my stay in IISc more cherishable. I would also like to thank Monu, Chidansh, Ramasya, Gaurav, Mangesh and many others from my seniors' batch, who gave useful advice about the project before it started.

Finally, and most importantly, I thank my parents, sister and other close relatives for their love and faith in me. Without their emotional support I would not have accomplished anything.

Abstract

RNA secondary structure prediction with pseudoknots is important, since pseudoknots are part of functionally improtant RNAs in cells. State of the art dynamic programming algorithms due to Akutsu et al [7] and Deogun et al [8] perform well on single RNA sequences. Our aim of this project is to be able to predict secondary structure of real life RNA sequences, which can be more than 700 nucleotides length. Recurrence for solving simple pseudoknots involve $O(n^4)$ time and $O(n^3)$ complexity. Existing dynamic programming algorithms solve this recurrence over full length of RNA sequence, which make them impractical for sequences of length greater than 500 nucleotides. We propose novel two-stage dynamic programming algorithm which can overcome this drawback by predicting regions of pseudoknots in first stage and predicting actual structure of pseudoknots in second stage. Another scalable approach HxMatch, *by Witwer et al [15] predicts secondary structure of consensus sequence of multiple aligned RNA sequences. HxMatch uses Maximum Weighted Matching (MWM) [30], which has $O(n^3)$ time complexity, for choosing basepairs based on score-matrix. Few observations suggest that score-matrix is so good that even simpler algorithms can be used for choosing basepairs rather than MWM. We propose to use Random Sampling algorithm with time complexity of $O(n^2 \log n)$ for choosing basepairs, instead of MWM. We performed experiments on Pseudobase dataset and other real life dataset taken from Rfam and NCBI. Two-stage algorithm outperforms both Deogun's algorithm and Rivas' algorithm in terms of time, however, prediction accuracy is less than Deogun's algorithm but better than Rivas' algorithm. Random Sampling algorithm performs as good as MWM for most of the sequences.*

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Most methods applied for RNA folding have been based on different heuristics such as quasi-Monte Carlo search, genetic algorithms, stochastic contextfree grammars, and the Hopfield networks, and dynamic programming algorithms. These approaches have limitations. For example, dynamic programming approaches, which are considered best [7] [8], have worst case time and space complexities of $O(n^4)$ and $O(n^3)$, respectively. Hence they are not appropriate for sequences of length greater than 500 nucleotides, considering time and space requirements involved. Also these approaches, perform on single RNA sequence.

Recurrence for simple pseudoknot prediction results into $O(n^4)$ time and $O(n^3)$ space complexities for Akutsu's algorithm. Their algorithm tries to solve this recurrence over full length. Prediction time can be improved by two-stage dynamic programming algorithm. In the first stage, algorithm predicts regions of pseudoknot and in second stage, it predicts actual structure of pseudoknot. Chapter 4 gives details about two-stage dynamic programming algorithm. Result comparison for Pseudobase dataset (section 4.4) shows that two-stage algorithm is quite faster compared to algorithms of Deogun et al and Rivas et al. For example, sequences of length 121 nucleotides can be predicted in 0.02 seconds by two-stage algorithm, whereas Deogun's algorithm takes 8 minutes and Rivas' algorithm takes around 4 hours 30 minutes. Deogun's algorithm was able to predict 95% pseudoknots and with 78% correct or almost correct predictions. Two-stage

algorithm predicts pseudoknot regions for 86% sequences, with 54% being correct or almost correct, which is still better than Rivas' algorithm, which could predict correct pseudoknots for only 50% sequences.

For longer RNA sequences, information from multiple alignment of same family of RNAs helps to define better score matrix, which in turn helps in prediction. Tabaska et al. [16] uses Maximum Weighted Matching (MWM) for predicting secondary structure of consensus sequence of multiple alignment. A matching in a graph is a collection of edges that pair-wisely do not have vertices in common. The predicted RNA structure is obtained as the matching that maximizes the sum of edge weights that are calculated from a combination of mutual information scores with helix scores for every possible base pair in a given multiple sequence alignment. Tabaska's helix score assigns a good pair score to Watson-Crick ($A = U, G \equiv C$) and $G = U$ pairs, a negative pair score to every other type of base pair and a penalty for gaps. Thus, it incorporates thermodynamic information (in a very simplified way) into the initial weight matrix. The MWM problem for any given weight matrix can be solved in $O(n^3)$ time and $O(n^2)$ space complexities , i.e., with the same effort as RNA folding problem for the pseudoknot free case. Tabaska's approach can not do well because of the quality of the initial weight matrix which often requires many sequences in the input alignment.

Witwer et al [15] proposed algorithm, called as *HxMatch*. HxMatch is scalable algorithm, as it takes only few seconds for predicting bisecondary structure of consensus sequence of multiple alignment of RNAs with length around 1500 nucleotides. HxMatch is different from Tabaska's approach in two manner. First, it uses more than just simplified energy score for initial weight matrix. Second, it postprocesses output by MWM to finally get structure, which they restrict to be bisecondary structure. *HxMatch* uses information from multiple alignment of RNA sequences to calculate score matrix for consensus RNA sequence of alignment. Then it chooses basepairs using MWM algorithm. This structure may not be a valid bisecondary structure. So it postprocess output of MWM to get valid bisecondary structure. Finally, it performs iterations of MWM and postprocess until convergence. MWM has time complexity of $O(n^3)$. Random sampling

of basepairs can be performed in $O(n^2 \log n)$ time complexity. We compare results of random sampling algorithm with MWM on real life datasets, taken from Rfam and NCBI. Result (section 5.4) shows that random sampling algorithm performs as good as MWM for most of the sequences.

This report is organized as follows. Chapter 2 gives brief detail about RNAs, its biological importance and its structural details. Chapter 3 discusses previous work done on RNA secondary structure prediction. Among the approaches discussed, dynamic programming algorithms and HxMatch [15] are considered best for RNA secondary structure prediction. Hence, more details are given about these algorithms, in particular. Chapter 4 gives details about two-stage dynamic programming algorithm including results comparison. HxMatch algorithm with random sampling for choosing basepairs is discussed in chapter 5.

Chapter 2

RNAs and Their Structure

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. RNAs contain four different nucleotides, Adenine (A), Guanine (G), Cytosine (C), and Uracil (U). The sequence of nucleotides of an RNA molecule constitutes its primary structure, and the pattern of pairing between nucleotides determines the secondary structure of an RNA.

2.1 Biological Importance

Different RNAs can play different roles in cells. For example, an RNA can function in the transmission and storage of genetic information, and they may also have structural and catalytic roles. RNA serves as the genetic material of many viruses. In all cellular organisms, information stored in DNA is used to govern cellular activities through the formation of RNA messages. Thus, similar to DNA, messenger RNAs (mRNA) can carry information. Ribosomal RNAs (rRNA) serve as structural scaffolds on which the proteins of the ribosome can be attached and as elements that recognize and bind various soluble components required for protein synthesis. Another example of structural RNAs is that of small nuclear RNAs (snRNA). These form a vital part of sliceosomes that process mRNAs in eukaryotes. Some RNAs play an important role in many chemical reactions.

One of the ribosomal RNAs of the large subunit acts as the catalyst for the reaction by which amino acids are covalently joined during protein synthesis. Other RNAs are able to catalyze RNA processing. RNAs that have a catalytic role are called ribozymes.

RNA molecules are continuous single strands that when fold back on themselves they form secondary structures consisting of extensive double stranded segments of variable length and complex tertiary structures. The double-stranded regions are held together by hydrogen bonds between the bases. This is the same principle that is responsible for forming the double helix of DNA molecules. Watson-Crick (WC) ($A = U, G \equiv C$), wobble ($G = U$) and other, non-canonical pairings can occur when the RNA is folded. Base triples are possible but rare. Many RNAs fold into structures that have been shown to be important for regulatory, catalytic, or structural roles in a cell. For example, secondary structure explains in part translational control in mRNA [10] and replication control in single-stranded RNA viruses [11]. Some RNAs do not code for proteins or structural RNAs [12] [13] and it is likely that the secondary structure of such transcripts defines their regulatory function in the cell. Thus, the knowledge of secondary and tertiary structures of an RNA molecule is highly desirable when investigating its role in a cell.

In recent years, biophysical techniques have been developed to determine the tertiary structures of small RNAs, and it has become evident that RNA molecules can achieve a level of structural complexity approaching that of proteins. Current biophysical methods to determine RNA structures are extremely time consuming and expensive. Therefore, bioinformatics approaches for accurate prediction are highly desirable.

2.2 RNA Structure

2.2.1 Secondary Structure

A *primary structure*, A, is RNA sequence of length n. $A = a_1a_2a_3 \ldots a_n$, where $a_i \in \{A, U, G, C\}$. Primary structure of RNA is string over nucleotides $\{A, U, G, C\}$.

A *secondary structure*, S, on an RNA sequence, $A = a_1a_2a_3 \ldots a_n$, is a set of base

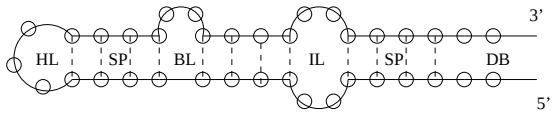


Figure 2.1: Schematic representation of the most relevant simple RNA secondary structures. The nucleotides are represented by circles.

pairs. A base pair between nucleotides a_i and $a_j, (i < j)$ is denoted by (a_i, a_j) or simply by (i, j). Base triples are quite rare. Sharp turns are prohibited. A turn, called a hairpin loop, must contain at least 3 bases.

Let, set of base pairs over sequence $a_I a_{I+1} \dots a_K, 1 \leq I < K \leq n$ be,

$$M_{I,K} = \{(i,j) | I \leq i < j \leq K, (a_i, a_j) \quad \text{is a base pair and every } i \text{ and } j \text{ appears at most once}\} \tag{2.1}$$

Then $M_{I,K}$ is called an RNA secondary structure without pseudoknots if no distinct pairs $(a_i, a_j), (a_h, a_k) \in M_{I,K}$ satisfy $i \leq h \leq j \leq k$ (Figure 2.2.A).

RNA secondary structure forms various loop regions in addition of pseudoknots and dangling bases (Figure 2.1). A hairpin loop (HL) is a sequences of unpaired bases bounded by one base pair. A stacking pair (SP), a bulge loop (BL) and an internal loop (IL) are all bounded by two base pairs. In a stack, the two base pairs are contiguous at both ends. In bulge, the two base pairs contiguous at one end. In an internal loop, the two base pairs are not contiguous at either ends. A dangling base (DB) is a single stranded base adjacent to $5'$ and $3'$ end of a base pair. More over, secondary structure can contain multi loops (ML), in which loop is bounded by more than two base pairs.

2.2.2 Pseudoknots

A set of base pairs $M_{I,K}$ is a *pseudoknots* (Figure 2.2.B) if there exist positions j' and j'', where $(I < j' < j'' < K)$, such that each pair $(i, j) \in M_{I,K}$ satisfies either $I \leq i < j' \leq j < j''$ or $j' \leq i < j'' \leq j \leq K$, and if pairs (i, j) and $(i_1, j_1) \in M_{I,K}$ satisfy either $i < i_1 < j'$ or $j' \leq i < i_1$, then $j > j_1$ holds.

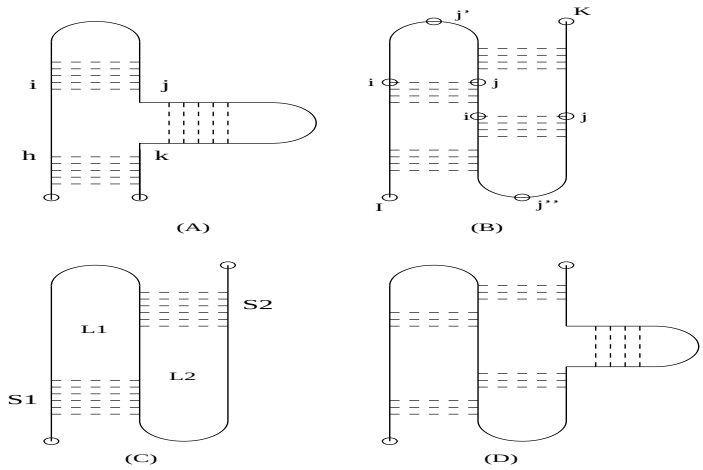


Figure 2.2: Examples of RNA secondary structure with pseudoknots.

The simplest pseudoknot, the classical or so-called *H-(hairpin) pseudoknot* contains two stems (S_1 and S_2) and two loops (L_1 and L_2) (Figure 2.2.C). Such pseudoknot is formed by the pairing of a hairpin loop, closed by S_1, with the downstream nucleotides forming S_2, or alternatively, by pairing of a hairpin loop, closed by S_2, with the upstream nucleotides forming S_1. Generally, pseudoknots can have *recursive* structure where any loop region can be replaced by another secondary structure with or without pseudoknots (Figure 2.2.D). Akustu [7] proved that RNA secondary structure prediction with generalized pseudoknots is NP-hard, when an arbitrary energy function depending on adjacent base pairs is used.

2.2.3 Bisecondary Structure

A *bisecondary structure* can be seen as super-position of two disjoint nested secondary structure. By nested, we mean for every 2 distinct basepairs (i, j) and (h, k) where $(i < j)$ and $(h < k)$, if $i < h < j$ then $i < k < j$ must hold. See figure 2.3.

The class of pseudoknots that can be predicted by dynamic programming algorithms is often given implicitly by the recursions of the algorithm. For example, class of pseudoknots addressed by Akutsu [7] algorithm are subset of class of pseudoknots addressed by Rivas and Eddy [6] (R&E class) algorithm. By the definition of *recursive pseudoknots*,

(a) Upper nested structure (b) Lower nested structure (c) Superposing upper and lower nested structures

Figure 2.3: Bisecondary Structure.

which addresses Akutsu's class of pseudoknots, it follows that bisecondary structure are superset of Akustu's class of pseudoknots. However, the R&E class is neither a subset of bisecondary structures since there are non-bisecondary structures contained (e.g., α-operon mRNA structure) in it, nor are bisecondary structures a subset of the R&E class since there are bisecondary structures not contained in the R&E class. Almost all known RNA pseudoknots fall into the class of bisecondary structures. Pseudobase contains over 245 examples of pseudoknotted structures which are all bisecondary structures, with the single exception of the Escherichia coli α-operon mRNA. Thus, by predicting bisecondary structure or RNA sequence, some pseudoknots which do not belong to class of simple or recursive pseudoknots, can be predicted.

Chapter 3

Computational Approaches for RNA Secondary Structure Prediction

Current biophysical methods to determine RNA structures are extremely time consuming and expensive. Therefore, bioinformatics approaches for accurate prediction are highly desirable. This chapter gives brief detail about few existing algorithms for RNA secondary structure prediction, which are related to our work. Dynamic programming algorithm and HxMatch are discussed in more details in following sections.

3.1 Previous Work

Several studies have been conducted, both from a practical and theoretical viewpoints to predict RNA secondary structure with pseudoknots. Most methods, which are capable of predicting pseudoknots adopt different heuristic search procedures, none of which is guaranteed to find an optimal structure. Moreover, current approaches fail to determine how far a given prediction is from optimality given a thermodynamic model. Such approaches include quasi-Monte Carlo search, genetic algorithms, methods based on an extension of the stochastic contest-free grammars developed by Brown and Wilson [4] and on the Hopfield network developed by Akiyama and Kanehisa [3].

Waterman et al [2] were the first to introduce a dynamic programming algorithm

for RNA secondary structure prediction (WS algorithm). Their algorithm considers recurrence for hairpin loop (HL), stacking pairs (SP), bulges (BL) and internal loops (IL). Hairpin loop and stacking pairs can be solved in $O(n^2)$ time and bulges can be solved in $O(n^3)$ time. They showed that intuitive recurrence for solving internal loop, which is of $O(n^4)$ time complexity, can be reformulated to make it of $O(n^3)$ time complexity. They showed that for internal loop bounded by basepairs (i, j) and (k, l) , $(i+1 < k, j+1 < l)$ recurrence can be solved for fixed (i, j). Solving it for all k's and l's results into $O(n^4)$ time complexity. But they showed that for constant $(l - k)$, score remains the same. So they reformulated recurrence in terms of $(l - k)$ to make it of $O(n^3)$ time complexity. However, they did not mention anything about pseudoknots. Their algorithm was suited only for non-pseudoknotted structure prediction.

WS algorithm first creates score-matrix h, where $h_{ij}, i < j$ is set to energy value, if i^{th} and j^{th} nucleotides can form basepair, otherwise its set to $+\infty$. Since energy values are used as score function, algorithm tries to minimize score. It writes RNA sequence in reverse order along the column of score-matrix, which makes more sense since RNAs fold onto itself to form secondary structure (see Waterman-Smith [2] for more details). This lead us to intuition that if A's and U's are interchanged and similarly if G's and C's are interchanged in the reverse RNA sequence, then we get modified sequence for same RNA. With this two sequences - original RNA sequence and modified sequence - helices can be seen as common subsequence between two sequences or in the score-matrix, it can be seen as diagonals. With this intuition, secondary structure prediction problem can be seen as finding maximum subsequence, that too with higher length or finding maximum diagonals with higher length in score-matrix. This intuition finally lead us to two-stage dynamic programming algorithm (chapter 4).

Rivas and Eddy [6] developed a dynamic programming algorithm for predicting optimal (minimum energy) RNA secondary structure, 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 [9]. The description of the algorithm is complex for both nested and non-nested configurations. The key point of their pseudoknot algorithm is the use of one-hole or gap matrices as a generalization of the matrices required for nested configuration.

Uemura et al. [5] proposed an algorithm based on tree adjoining grammar(TAG). The time complexities of their algorithm depends on 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 [7] analyzed their method and found that tree adjoining grammar was not crucial but the parsing procedure was crucial. Since the parsing procedure is intrinsically a dynamic programming procedure, Akutsu [7] reformulated their method as a dynamic programming procedure without tree adjoining grammar.

Deogun et al. [8] implemented and analyzed the performance of a dynamic programming algorithm following Akutsu's for predicting simple pseudoknots in optimal secondary structure of a single RNA sequence using standard thermodynamic parameters for RNA folding. The algorithm has the worst case time and space complexities $O(n^4)$ and $O(n^3)$, respectively. They developed implementation for algorithm to find minimal energy RNA structures using the current RNA structure thermodynamic model [14]. They validated the accuracy of algorithm on the entire simple pseudoknot collection in the PseudoBase.

To best of our knowledge, Deogun et al. [8] has reported best accuracy. Considering base pairs score, their dynamic programming algorithm is same as Akutsu's [7]. Their algorithm is not practical for sequences of length greater than 500. Section 4.4 compares results of two-stage algorithm with Deogun's results.

Witwer et al. [15] proposed HxMatch algorithm. HxMatch uses information available from multiple alignment of RNA sequences from same family of RNAs, to define better score matrix. It uses MWM algorithm to chose basepairs for RNA secondary structure. This, however, may result into non-bisecondary structure. Hence it postprocesses output

of MWM to finally get bisecondary structure. HxMatch performs iterations of MWM and postprocess until convergence. HxMatch can predict for sequences of length around 1500 nucleotides within few seconds. However, it uses MWM to chose basepairs, which is $O(n^3)$ algorithm. Chapter 5 shows that random sampling can be used instead of MWM to chose basepairs. Random sampling has $O(n^2 \log n)$ time complexity. It performs as good as MWM algorithm for most sequences.

3.2 Dynamic Programming Approaches

Aktusu [7] proposed dynamic programming algorithm, for finding optimal RNA secondary structure containing simple pseudoknots with maximum base pairs score. Algorithm considers only watson-crick($A = U, G \equiv U$) and wobble ($G = U$) basepairings.

Let S be score-matrix. $S(i,j)$ is basepairs score between i^{th} and j^{th} nucleotides of RNA sequence. $S_{nonpk}(i,j)$ is basepairs score between i^{th} and j^{th} nucleotides of RNA sequence provided it does not contain pseudoknot. $S_{pk}(i,j)$ is basepairs score between i^{th} and j^{th} nucleotides of RNA sequence, containing simple pseudoknot.

3.2.1 Non-pseudoknotted Structure

S_{nonpk} is the score-matrix for RNA secondary structure without pseudoknots. $S_{nonpk}(i,j)$ is calculated as,

$$S_{nonpk}(i,j) = \max \begin{cases} S_{nonpk}(i+1, j-1) + V_{i,j} \\ \max\limits_{i \le k < j} \{S_{nonpk}(i, k-1) + S_{nonpk}(k, j)\} \end{cases} \tag{3.1}$$

where $V_{i,j}$ is set to 1 if i^{th} and j^{th} nucleotides can form a basepair. Hence,

$$V_{i,j} = \begin{cases} 1 & (a_i, a_j) \in \{(A,U), (G,C), (G,U)\} \\ 1 & (a_i, a_j) \in \{(U,A), (C,G), (U,G)\} \\ 0 & otherwise \end{cases} \tag{3.2}$$

3.2.2 Pseudoknotted Structure

Dynamic programming algorithm to find RNA secondary structure with simple pseudoknots requires, three 3-dimensional $(n \times n \times n)$ matrices called, SL, SM, SR and one 2-dimensional matrix $(n \times n)$ called S_{pk}. Here n is the length of RNA sequence. For finding a simple pseudoknot substructure whose endpoints are I^{th} and K^{th} residues, the algorithm considers three types of triplets $SL(i,j,k), SM(i,j,k)$, and $SR(i,j,k)$ for each i, j, and k such that $(I \leq i < j < k \leq K)$.

SL, SM, SR matrices are defined in the following way: $SL(i,j,k)$ is the score of the best folding between positions I and i, and j and k, provided that i^{th} and j^{th} residues make a base pair; $SR(i,j,k)$ is the score of the best folding between positions I and i, and j and k, provided that j^{th} and k^{th} residues make a base pair; $SM(i,j,k)$ is the score of the best folding between positions I and i, and j and k, provided that neither i pairs with j, nor j pairs with k.

$S_{pk}(I,K)$ is the score of the best pseudoknot fold with ending points I and K. Complete recurrence relations for computing these triplet can be found in [7]. For each pair (I,K) such that $I+6 < K$ algorithm computes the above three matrices and obtain the score of a pseudoknot $S_{pk}(I,K)$ with endpoints I and K by

$$S_{pk}(I,K) = \max_{I<i<K} \begin{cases} SL(i,i+1,K) \\ SM(i,i+1,K) \\ SR(i,i+1,K) \end{cases}$$

Finally, the optimal score for each pair of $(i,j), 1 \leq i < j \leq n$ can be computed by the following recurrence:

$$S(i,j) = \max \begin{cases} S_{pk}(i,j) \\ S_{nonpk}(i+1,j-1) + V_{i,j} \\ \max\limits_{i \leq k < j} \{S_{nonpk}(i,k-1) + S_{nonpk}(k,j)\} \end{cases} \tag{3.3}$$

,where $V_{i,j}$ is defined in Equ. 3.2.

To overcome this drawback of algorithm, we propose two-stage dynamic programming algorithm (Chapter 4) for RNA folding. In the first pass, our algorithm predicts candidate regions for simple pseudoknots. After getting these regions it predicts actual structure of these regions. Thus, success of two-stage dynamic programming algorithm is largely dependent on first pass. First pass should quickly and accurately predict pseudoknot regions, so that search space is reduced for second pass.

3.2.3 Time and Space Complexities

Rivas' algorithm has worst case time and space complexities of $O(n^6)$ and $O(n^4)$ respectively. Akutsu's algorithm has worst case time and space complexities of $O(n^4)$ and $O(n^3)$ respectively. Recurrence for non-pseudoknotted structure (section 3.2.1) can be solved in $O(n^3)$ time complexity. Recurrence for solving 3-dimensional matrix SL, SM, SR (section 3.2.2), which are being used for predicting simple pseudoknots, results into time and space complexities of $O(n^4)$ and $O(n^3)$. This recurrence makes it infeasible for sequences of length greater than 500 nucleotides. It unnecessarily explores full RNA sequence for pseudoknots, since simple pseudoknots generally will not occur over full sequence. For example, for sequences of length 121 nucleotides, Deogun's algorithm takes 8 minutes and Rivas' algorithm takes 4 hours 30 minutes.

3.3 HxMatch Algorithm

The hxmatch algorithm starts from a multiple alignment and generates a scoring matrix that assigns a weight to each possible basepair. This yields a weighted graph $\Gamma^{(0)}$, where the nucleotides form the vertex set and the edge set contains all base pairs with positive weight. In the next step, an MWM algorithm finds the matching on $\Gamma^{(0)}$ that maximizes the sum of the edge weights. The base pairs contained in the matching include isolated base pairs and do not necessarily form a bisecondary structure. Therefore, the maximum matching needs to be postprocessed. During postprocessing, several edges are deleted from the original input graph resulting in a modified weighted graph $\Gamma^{(1)}$. The

computation of the maximum matching and postprocessing are iterated to convergence. The crucial part of hxmatch is the improved scoring procedure which is describe in detail in the following subsections.

3.3.1 Basepair Scoring

Starting from a RNA sequence alignment $\mathcal{A}$ of N sequences, a scoring matrix Π is generated from the combination of the thermodynamic score, derived from the stacking energies of helices, and the covariation score, which is based on the number of mutations for a given alignment position.

Thermodynamic score: For each sequence $\alpha \in \mathcal{A}$, all basepairs (i, j) contained in the set of allowed basepairs $\beta = \{GC, CG, AU, UA, GU, UG\}$ which are part of a possible helix with minimum length 3 are tabulated. The energy of each helix is calculated using the (experimentally determined) standard energy model for thermodynamic RNA folding [17]. The weight H_{ij}^{α} of a basepair in sequence α is the energy of the longest helix the base pair is part of, multiplied by (-1) to obtain positive weights. The entry in the combined scoring matrix $H_{ij}^{\mathcal{A}}$ of the alignment is then

$$H_{ij}^{\mathcal{A}} = \frac{1}{N} \sum_{\alpha \in \mathcal{A}} H_{ij}^{\alpha} \tag{3.4}$$

Covariation score: Hxmatch uses a covariance score instead of the mutual information scores [18] preferred by many authors. The reason is that mutual information measures do not make explicit use of the RNA base-pairing rules. While this allows the identification of noncanonical base pairs and tertiary interactions it is less sensitive to information that supports conserved helices: consistent, noncompensatory mutations, in which only one side of a base pair is mutated, e.g., GC to GU, yield a score of 0 just as GC to GA mutations. The covariance score

$$C_{ij} = \sum_{XY, X'Y'} f_{ij}(XY) D_{XY,X'Y'} f_{ij}(X'Y') \tag{3.5}$$

Parameter	Default
ϕ_1	0.8
ϕ_2	60 kcal/mol
A	75 kcal/mol
p_{min}	0.90
Π^*	25 kcal/mol

Table 3.1: Default parameter values

was introduced in [19]. Here, $f_{ij}(XY)$ denotes the frequency of a pair of type XY at positions i and j of the alignment $\mathcal{A}$. The 16×16 matrix D has entries,

$$D_{XY,X'Y'} = \begin{cases} 0 & \text{if } XY = X'Y' \text{ or } XY \notin \beta \text{ or } X'Y' \notin \beta \\ 1 & \text{if } XY, X'Y' \in \beta \text{ , } X = X' \text{ or } Y = Y' \\ 2 & \text{if } XY, X'Y' \in \beta \text{ , } X \neq X' \text{ , } Y \neq Y' \end{cases} \tag{3.6}$$

While consistent mutations add to the weight of a base pair, nonconsistent mutations incur a penalty. Let q_{ij} be the fraction of inconsistent sequences for positions i and j, i.e., sequences that cannot form a base pair between positions i and j. They are taken into account by forming the combined score,

$$B_{ij} = C_{ij} - \phi_1 q_{ij} \tag{3.7}$$

Together with the helix score, the combined weight,

$$\pi_{ij} = H_{ij}^{\mathcal{A}} + \phi_2 B_{ij} \tag{3.8}$$

where ϕ_1 and ϕ_2 are scaling factors, their default values are given in 3.1. Note that ϕ_2 has the dimension of an energy and is given in kcal/mol.

The MWM algorithm does not account for any dependencies between base pairs. Therefore, dependencies between base pairs, i.e., the formation of helices, have to be

reflected by the score of each base pair. Therefore, hxmatch does not use π_{ij} itself, but rather include an additional aggregation step. It determines all maximal stem-loop structures Ψ consisting of helices of length at least 3 and bulges with a single base pair which consist of base pairs with positive weight π_{ij}. The weight of the stem-loop structure Ψ is the sum of the weights of its base pairs:

$$\omega_\Psi = \sum_{ij \in \Psi} \pi_{ij} \tag{3.9}$$

Finally, hxmatch assigns to each base pair (i, j) the weight ${\Pi'}_{ij}$ of the stem-loop structure with the largest weight that passes through it: ${\Pi'}_{ij} = \omega_\Psi$ for all $ij \in \Psi$ with $\pi_{ij} > 0$. This strongly favors base pairs that are part of longer helices, the weight is essentially proportional to the square of helix length. This score compensates the tendency of the MWM algorithm to produce many short helices.

Energy-based prediction methods tend to predict long range base pairs much less reliably [20] [21] and predicted long range pairs account for many of the false positives. In high-quality structures determined by comparative analysis, 75 percent of the base pairs span less than 100 nucleotides [20] [22] . Furthermore, the stacking energies of long-range helices of natural RNAs increase with the range of the helix [23] [24], i.e., where the functional structure of an RNA molecule requires long-range pairs, evolution selects unusually stable helices. A likely explanation is that short range pairs are favored by the folding kinetics [25]. To avoid producing too many long range pairs, a penalty is applied to base pairs spanning more than 400 nucleotides,

$${\Pi''}_{ij} = \begin{cases} {\Pi'}_{ij} - 0.1(j - i - 400) & \text{if } 400 < j - i < 800 \\ {\Pi'}_{ij} - 40 & \text{if } j - i \geq 800. \end{cases} \tag{3.10}$$

This penalty function was determined empirically.

It is easy to take into account scores from other sources. For example, RNAalifold [19], which is part of the Vienna RNA Package [26] [27] calculates the consensus secondary structure without pseudoknots for a set of aligned sequences, furthermore, it

calculates the base pairing probabilities in addition to the minimum free energy structure. RNAalifold takes into account phylogenetic information by adding a covariance score to the energy function of the standard energy model [17] [28]. Hxmatch provides the option $-A$ for assigning a bonus A to all base pairs contained in the RNAalifold prediction, option $-AP$ assigns a bonus $AP = 2A \times ln(p/p_{min})$ to all base pairs with base pairing probability p exceeding a threshold p_{min}, refer to Table 1.

Finally, all base pairs with a score smaller than a threshold Π^* get zero weight. The resulting final weights Π_{ij} are then used for the MWM computation.

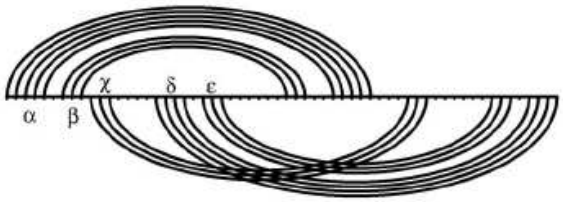
All parameters have been empirically optimized, their default values are given in Table 1. The value of ϕ_2 scales the covariation score so that the ratio of the range covered by the covariation score to the range covered by the thermodynamic score is approximately 3:1. The value of Π^* is in the order of magnitude of 5 percent of the maximum weight.

3.3.2 Predicting Bisecondary Structure

After calculating score-matrix, HxMatch performs MWM to choose basepairs. The input graph $\Gamma^{(0)}$ for the maximum weighted matching algorithm consists of the vertex set $V = 1, 2, \ldots, n$ where n is the length of the alignment and the edge set formed by all base pairs with score $\Pi_{ij} > 0$. Hxmatch uses the algorithm for maximum weighted matching of Gabow [30] implemented by Rothberg [31].

The maximum weighted matching obtained for the input graph $\Gamma^{(0)}$ is not necessarily a bisecondary structure. Furthermore, isolated base pairs are contained in the matching. Therefore, the outcome of the MWM algorithm needs some postprocessing. All isolated base pairs and helices with length 2 are deleted from the outcome and the remaining helices are extended further, if the corresponding base pairs are contained in the graph $\Gamma^{(0)}$.

Hxmatch uses the following greedy procedure to derive a bisecondary structure from the matching. The helices are ordered by descending weight. Initially, all helices are assigned to Ω_U, the subset of helices which are drawn in the upper half plane of the



Helix	Weight	assigned to
α	100	Ω_U
δ	90	Ω_L
ϵ	70	Ω_L
χ	65	deleted
β	45	Ω_U

Figure 3.1: Classification of helices: Since helix χ is inconsistent with the higher ranked helix $\alpha \in \Omega_U$ and helix $\delta \in \Omega_L$, it is deleted to obtain a bisecondary structure.

linked diagram representation (see Figure 3.1). Then, it goes through the sorted list of helices and assign all helices conflicting with a higher ranked helix (temporarily) to Ω_L. Subsequently, the helices contained in Ω_L are scanned and all helices conflicting with a higher ranked helix of Ω_L are deleted from the graph. Figure 3.1 shows an example of the classification of the helices.

Hxmatch then removes from the original graph $\Gamma^{(0)}$ all base pairs conflicting with the bisecondary structure predicted in the first round. This yields a modified graph $\Gamma^{(1)}$, which serves as input for a second run of the maximum weighted matching algorithm. This allows additional base pairs to be added to the bisecondary structure of the previous run, but may again yield a nonbisecondary structure. Therefore, the two steps (MWM and postprocessing) are iterated until the outcome stays constant. For the data sets investigated, at most, four iterations were needed.

3.3.3 Time and Space Complexities

In case of Hxmatch, tabulating all possible helices for the individual sequences requires $O(Nn^2)$ time and $O(n^2)$ memory, with N being the number of sequences and n being the length of the alignment. Scanning the combined helix score for helices allowing bulges of size one, requires less than $O(n^3)$ time since helix lengths are (almost) independent of n [32] [33] and the mean number of alternatively helices a base pair is part of is small, in

practice. The MWM algorithm requires $O(n^3)$ time and $O(n^2)$ memory. Since $N \ll n$, the overall complexity is $O(n^3)$ time and $O(n^2)$ memory. RNAalifold is also $O(n^3)$ in time and $O(n^2)$ in memory. The *hxmatch* program in combination with RNAalifold needs only seconds for the structure prediction of a 16SrRNA on a Linux PC with a Dual XEON P4 2.2 Ghz. For comparison, ilm [34] takes about five minutes for the same task.

Dynamic programming algorithm for RNA secondary structure prediction by Akutsu et al [7] performs good in terms of prediction accuracy, but are not suitable for sequences of length greater than 500 nucleotides due to its $O(n^4)$ time complexity for recurrence of simple pseudoknot prediction. Two-stage dynamic programming algorithm can be used to improve time, where in the first stage only regions of pseudoknot are predicted to reduce search space. Next chapter gives more details about two-stage dynamic programming algorithm with results comparisons.

Chapter 4

Two-stage Dynamic Programming Algorithm

In this chapter, we propose a dynamic programming algorithm for finding regions of pseudoknots for given RNA sequence (first stage). Also, performance of two-stage algorithm is compared with Deogun et al [8] and Rivas and Eddy [6]. As described in (Section 3.2), success of overall algorithm is largely dependent on first stage of algorithm.

Based on the intuitions similar to Waterman and Smith algorithm [2] we make an assumption that, if base pair (i, j) appears in secondary structure and $V_{i+1,j-1} = 1$, then base pair $(i+1, j-1)$ will also appear in secondary structure. Matrix V is defined as in Equ 3.2. Based on this assumption, we define,

$$L_{i,j} = \left(1 + \arg\max_{k \geq 0}\{V_{i+k,j-k} = 1\}\right) 1_{\{V_{i-1,j+1}=0\}}$$

where $1_{\{y\}} = 1$ if y is true, 0 otherwise.

RNA sequence between i^{th} and j^{th} nucleotides is predicted as region of simple pseudoknot, if heuristic score of pseudoknotted structure exceeds score of non-pseudoknotted structure. If this task is done efficiently, significant amount of time can be saved. Pseudoknots generally occur over length of 100 nucleotides rather than over full length of RNAs which can be around 1500 nucleotides. Once these regions of pseudoknots (of

length around 100) can be predicted, algorithm can predict actual structure of pseudoknots. Thus, complexity of predicting simple pseudoknots is still $O(n^4)$, but unlike existing dynamic programming algorithm for which n can be around 1500, for two-stage dynamic programming algorithm n takes value of around 100.

4.1 Finding Score for Non-pseudoknotted Structure

Non-pseudoknotted RNA secondary structure with maximum base pairs score can be found out by a simple dynamic programming algorithm [1].

$$S(i,j) = \max \begin{cases} S(i+1, j-1) + V_{i,j} \\ \max\limits_{i<k\leq j} \{S(i,k-1) + S(k,j)\} \end{cases}$$

Initialization is done with $S(i,j) = 0$ for all $i \geq j$. Aim is to calculate $S(1,n)$. This algorithm has time and space complexities of $O(n^3)$ and $O(n^2)$ respectively. Optimal non-pseudoknotted structure can be found out from $S(i,j)$ using traceback technique [1] with same complexities. Also, this algorithm can be modified to consider minimal energy score functions.

4.2 Finding Score for Simple Pseudoknot Structure

$PS(I,K)$ is numbers of base pairs over sequence $a_I a_{I+1} \ldots a_K$ when $M_{I,K}$ contains simple pseudoknots. Let $PKEnd(I,j,K)$ be set of possible basepairs, which can be at start of stack pairs forming pseudoknot structure with stack pair starting at $(I,j), I < j < K$. Thus

$$\begin{aligned} &PKEnd(I,j,K) \\ &\quad = \{(h,K) | (I+l_1) < h < (j-l_1-l_2+2), \\ &\qquad l_2 < (K-j), where \quad l_1 = L_{I,j}, l_2 = L_{h,K}\} \end{aligned}$$

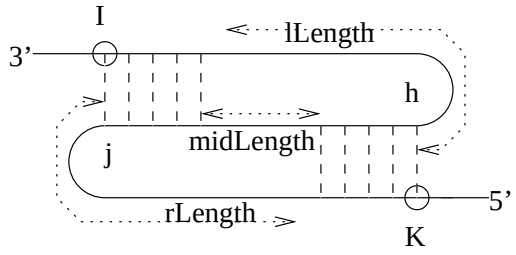


Figure 4.1: Schematic diagram for $H_{probable}$ calculation.

Start with initialization, $PS(I,K) = 0, (1 \leq I, K \leq n, I + 6 > K)$.

$$PS(I,K) = \max \begin{cases} PS(I, K-1) \\ PS(I+1, K) \\ PKScore(I,K) \end{cases}$$

Where

$$\begin{aligned} &PKScore(I,K) = \\ &\max_{I<j<K} \left\{ L_{I,j} + \max_{h \in PKEnd(I,j,K)} \{L_{h,K} + H_{probable}\} \right\} \end{aligned} \tag{4.1}$$

,where $H_{probable}$ is additional heuristic score for simple pseudoknot, whose stack pair ends are starting from (I,j) and (h,K), with stack pair lengths of $L_{I,j}$ and $L_{h,K}$ respectively (Figure 4.1). Let

$$\begin{aligned} lLength &= h - I - L_{I,j} \\ rLength &= K - j - L_{h,K} \\ midLength &= j - h - L_{I,j} - L_{h,K} \quad then \\ H_{probable} &= \min\{lLength + rLength, midLength\} \end{aligned} \tag{4.2}$$

As can be seen, in (Figure 2.1.B) for simple pseudoknots, all stack pairs other than two end stack pairs must have one base of base pairs between $(h + L_{h,K} - 1, j - L_{I,j} + 1)$ region and other base either in $(I + L_{I,j}, h)$ region or $(j, K - L_{h,K})$ region. Hence given two stack pairs, as an end stack pairs of pseudoknots, starting from (I,j) and (h,K),

then base pairs score is bounded by $PKScore(I, K)$ (Equ. 4.1) involving $H_{probable}$ (Equ. 4.2) term.

In fact, $H_{probable}$ can be set to zero for hairpin pseudoknots and for Recursive pseudoknots $H_{probable} = lLength + rLength + midLength$.

4.3 Finding Pseudoknot Regions

Pseudoknot regions over RNA sequence can be predicted using following algorithm.

$$S(I,K) = \max \begin{cases} PS(I,K) \\ S(I+1,K-1) + V_{I,K} \\ \max\limits_{I<j<K} \{S(I,j-1) + S(j,K)\} \end{cases}$$

This means that, whenever $PS(I, K)$ score is better than non-pseudoknotted score for region (I, K), $S(I, K) = PS(I, K)$. For other cases, always $PS(I, K) < S(I, K)$. Let CPK be set of all regions with higher score for pseudoknots than non-pseudoknotted structure. Thus,

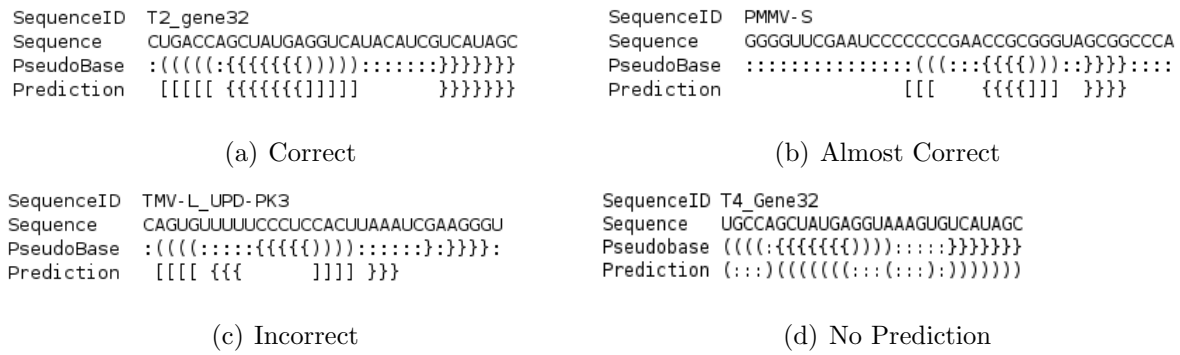
$$CPK = \{(l, r) | PKScore(l, r) = S(l, r)\}$$

4.4 Experimental Results and Evaluation

The algorithm was implemented in C++. We have implemented and evaluated for only first pass of algorithm. The program runs in a Unix environment on Intel Pentium-4 CPU 2.40 GHz, with 1 GB RAM. The input to program is RNA sequence consisting of string over A, U, G, C. The program outputs regions over sequence length, where there are good chances of finding pseudoknots. Actual structure prediction over the regions identified by first pass will be done in second pass. However, as we mentioned earlier, we do not have results for second pass.

To evaluate our approach, we use the set of RNA sequences from *PseudoBase* dataset, which is freely available at *http://biology.leidenuniv.nl/˜batenburg/PKB.html*. We found

189 sequences containing simple pseudoknots. The lengths of sequences vary from 21 to 121 nucleotides. Sequence structures are formatted like, ::(((:::{{{}))}}}:: . Since we are interested in regions of pseudoknots rather than structure, output only shows regions for which pseudoknot score is higher. Out of 189 sequences, our algorithm predicts pseudoknot regions for 163 sequences. Among those 163 pseudoknot regions, 103 regions were predicted correct or almost correct (Figure 4.2). Results are summarized in Table 4.1.



(a) Correct

(b) Almost Correct

(c) Incorrect

(d) No Prediction

Figure 4.2: Some typical outputs of the predictions. We are using [ ] instead of ( ), just to emphasis that stack pairs were selected due to $PKScore$ having higher value than non-pseudoknotted score.

Deogun et. al has reported accuracy [8] on the same dataset, PseudoBase. They reported results for 169 sequences. Their algorithm predicts pseudoknots for 163 sequences, out of which 131 sequences are predicted correct or almost correct. So their algorithm was able to predict 95% pseudoknots and with 78% correct or almost correct

Correct	39
Almost Correct	64
Incorrect	86
Total	189

Table 4.1: Result Summary. In case of, *almost correct*, predicted region is different by at most 4 position.

Conflict	4
$G = U$ Pairing	21
No Prediction	26
Superset	7
Other	28
Total Incorrect	86

Table 4.2: Breakup for Incorrect Pseudoknot Regions Prediction

predictions. Our algorithm predicts pseudoknot regions for 86% sequences, with 54% being correct or almost correct, which is still better than Rivas' algorithm, which could predict 50% sequences only.

However, our algorithm outperforms Rivas' algorithm and Akutsu's algorithm in terms of time and space requirements. Worst case time complexities are $O(n^{6.8})$ and $O(n^4)$ for Rivas' algorithm and Akutsu's (Deogun's) algorithm respectively. Our algorithm also has $O(n^4)$ time complexity, but since it operates over integer sparse matrix L only, significant speedup is achieved. For example, for sequence of 121 nucleotides, our algorithm takes 0.02 seconds. Deogun reported [8], for sequence length of 114 nucleotides, 8 minutes for their algorithm and 4 hours 30 minutes for Rivas' algorithm. Also space requirements for Rivas' algorithm and Akutsu's algorithm are $O(n^4)$ and $O(n^3)$ respectively, while our algorithm has $O(n^2)$ space complexity.

Table 4.2 lists breakup of incorrect pseudoknot regions prediction. *Conflict* means that for two end stack pairs of actual pseudoknot in regions (I, K), L matrix has corresponding entries for two stack pairs overlapping for one nucleotides. Because of this overlap our algorithm does not consider these two stack pairs for pseudoknot regions. As can be seen, ignoring $G = U$ pairings results in major prediction failure. In fact, ignoring $G = U$ pairings and *conflicts* result several times in *No Prediction*. *Superset* means that our algorithm predicted regions, which is superset of actual pseudoknot region. They account for increase in search space for second pass.

HxMatch algorithm by Witwer et al [15] can predict for sequences of length 1500

nucleotides within minute. It uses MWM for choosing basepairs after calculating score-matrix. Our observations suggests that simpler algorithms, for example random sampling, can be used instead of MWM for choosing basepairs. MMW has time complexity of $O(n^3)$ whereas random sampling can be performed with $O(n^2 \log n)$ time complexity. Next chapter gives more detail about our observations, random sampling algorithm and results comparison.

Chapter 5

Random Sampling Algorithm

As seen in section 3.3, HxMatch algorithm uses MWM algorithm to choose basepairs after calculating score-matrix. MWM algorithm has time complexity of $O(n^3)$. Section 5.1 shows that few basepairs get quite high weights compared to others. Such basepairs are having better chance of being in final secondary structure, compared to other basepairs with lower weights. Based on this observation, we propose to use random sampling for choosing basepairs, instead of MWM. Random sampling can be done in $O(n^2 \log n)$ time complexity. Also, as result section 5.4 shows, it performs as good as original HxMatch with MWM for most of the sequences.

5.1 Few Observations

In this section, we discuss few observations on Hxmatch algorithm and also about RNA sequences, in general.

5.1.1 About Hxmatch Algorithm

As seen in section 3.3, Hxmatch uses thermodynamic score and co-variation score to finally build score-matrix Π. Figure 5.1 is the plot of values of Π_{ij} for all basepairs $(i, j), \Pi_{ij} \geq 0$ with different scoring schemes. As can be seen, with different scoring scheme, actual value of Π_{ij} was differing but still comparative values among basepairs

remains the same. That means, if $\Pi_{ij} > \Pi_{hk}$ for some basepairs $(i,j),(h,k)$ with default scoring scheme then same was the case without considering co-variation. This may not hold true for all basepairs in general, but it looks like it holds for most of the basepairs. Hence equation 3.5 can be replaced with different cases.

Frequency of legal basepairs: In this variation of scoring scheme, instead of going for co-variation score, frequency of legal basepairs among all alignment sequences is considered. That means,

$$P_{ij}^1 = \sum_{XY} f_{ij}(XY)$$

$$\text{Hence} \quad B_{ij} = P_{ij}^1 - \phi_1 q_{ij}$$

We call this scoring scheme as P_{ij}^1.

All D matrix values set to 1: In this scoring scheme, values of D matrix are set to,

$$D_{XY} = \begin{cases} 1 & \text{if } XY \in \beta \\ 0 & \text{otherwise} \end{cases}$$

$$P_{ij}^2 = \sum_{XY,X'Y'} f_{ij}(XY) f_{ij}(X'Y')$$

$$\text{Hence} \quad B_{ij} = P_{ij}^2 - \phi_1 q_{ij}$$

We call this scoring scheme as P_{ij}^2.

No Co-variation: In this scoring scheme, co-variation score is ignored while calculating score-matrix. That means only thermodynamic score is considered for initial score-matrix. In this case,

$$\pi_{ij} = H_{ij}^{\mathcal{A}}$$

We call this scoring scheme as *NoCovar*.

As can be seen in table 5.1 there is not much difference between performances of *default Hxmatch* algorithm and *NoCovar*. Also, variation P_{ij}^2 (setting all D matrix values to 1) performs as good as *default Hxmatch* in most of the cases.

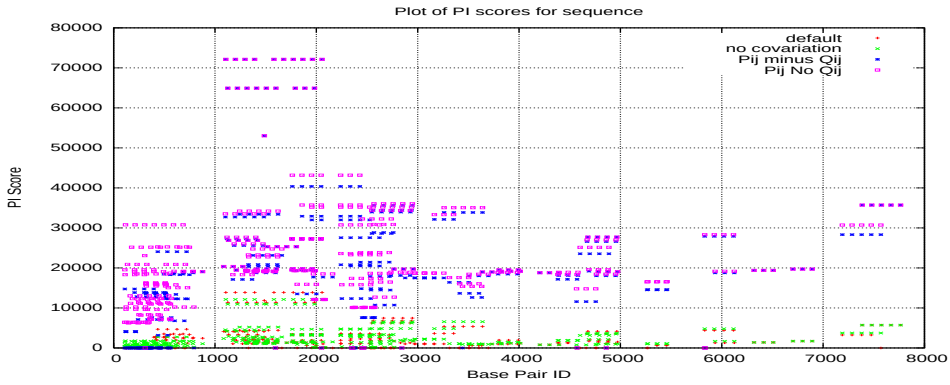


Figure 5.1: Values of Π_{ij} for basepairs (i, j) for RNA Tombusvirus.

	Default		NoCovar		P_{ij}^1		P_{ij}^2	
	SS	SP	SS	SP	SS	SP	SS	SP
Tombusvirus	37.5	39.1	37.5	40.9	58.3	37.5	56.0	39.1
Enterovirus	76.3	87.9	81.6	88.6	63.2	72.7	76.3	87.9
SRP RNA	91.9	86.8	83.7	80.9	59.3	71.8	91.9	86.8
tmRNA	80.2	90.4	70.8	78.9	67.0	66.4	67.0	79.8
Rnase P RNA	77.4	88.1	71.0	87.1	64.5	80.8	74.2	83.6
16S rRNA	77.6	85.7	71.5	74.7	45.8	57.5	77.6	85.7

Table 5.1: Results comparison for various scores. *Default* is Hxmatch with no changes.

5.1.2 Few observations about RNA sequences

For RNA sequence of length n, number of basepairs are bounded by $0.5n$. Also it has been observed that typically number of basepairs in secondary structures are between $0.25n$ to $0.4n$.

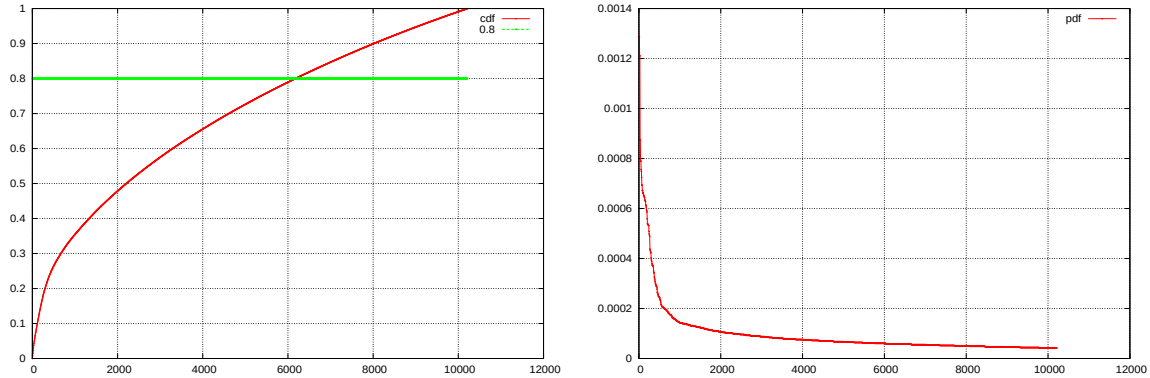
After sorting all possible basepairs with positive scores, i.e. $(i,j), \Pi_{ij} \geq 0$ in descending order based on their weights Π_{ij}, if we divide each such Π_{ij} by sum of all Π_{ij} score then it can be seen as *probability of basepair* (i,j) *being in secondary structure*. Let $Prob(i,j) = \frac{\Pi_{ij}}{\sum_{ij}\Pi_{ij}}$. It can be observed in figure 5.2 that, around 50% of basepairs account for 80% of cumulative weights. So one can consider only those basepairs which account for initial 80% of cumulative weights. As we show in result section 5.4 there is not much difference in accuracy with this assumption. In fact, same accuracy can be achieved with less number of iterations.

5.2 Random Sampling Algorithm

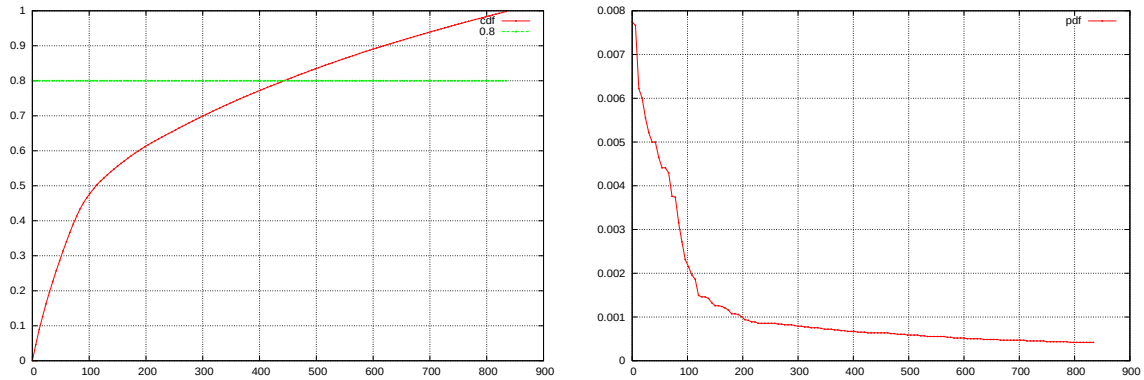
In this section, we show that instead of using MWM for choosing basepairs for structure prediction on score-matrix Π from Hxmatch, one can randomly sample basepairs. From the observations stated in section 5.1 (in particular figure 5.1 and figure 5.2) it is clear that scores of some basepairs are so high compared to other basepairs that they stand high chance of being in final secondary structure and then there are many basepairs with such a low scores (bottom 50% of basepairs which accounts for 20% of total weight, figure 5.2) that they may hardly be in final secondary structure.

This inspired us to try for simpler methods of choosing basepairs, given the score-matrix. One approach is to sample basepairs iteratively. In each iteration we calculate sum of weights of chosen basepairs. Finally we select iteration which gave maximum score. It can be basepairs score or energy score. Postprocessing is done on all basepairs chosen in that particular iteration to finally get bisecondary structure.

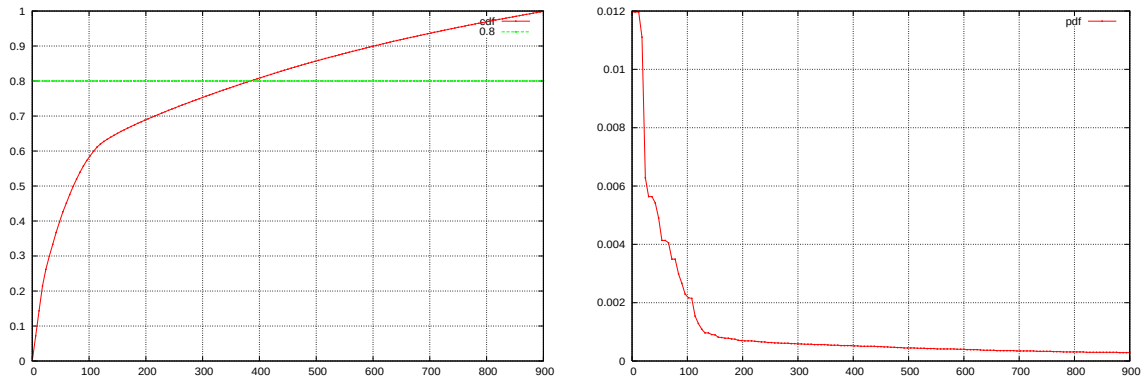
We choose basepairs in each iteration in following way. First of all we sort all basepairs (i,j) with $\Pi_{ij} \geq 0$ based on their weights Π_{ij}. Let $E = \{(i,j)|\Pi_{ij} \geq 0\}$



(a) 16S rRNA, length=1542nt



(b) RNase P RNA, length=404nt



(c) SRP RNA, length=305nt

Figure 5.2: CDF and PDF of weight of basepairs, taken as random variables. Basepairs are sorted by weights first.

be ordered set of all such basepairs. Probability of basepair (i, j) is calculated by $Prob(i, j) = \frac{\Pi_{ij}}{\sum_{ij \in E} \Pi_{ij}}$. However, in particular iteration of sampling basepairs, we choose basepair (i, j) only if no other basepair containing i or j nucleotide was chosen in any of the previous trials. That means, if in t^{th} trial of one iteration basepair (i, j) is picked randomly, then too it will be considered as *chosen* basepair only if no basepair from γ_i or γ_j was chosen in previous $t-1$ trials of same iteration. Here, $\gamma_i = \{(i, j) \in E\}$ is set of all edges containing i^{th} nucleotide.

Since number of basepairs in final structure is in $O(n)$, n being length of sequence, we decided to sample with constraints in multiples of n in each iteration, for example, $3n$ or $10n$ trials in each iteration. However, number of iterations and number of trials in each iteration needs to be fixed.

As seen in section 5.1 number of basepairs are bounded by $0.5n$ for any RNA. Hence instead of fixing number of trials in each iteration, we fixed number of basepairs chosen to be at most $0.4n$ in each iteration. Also we observed that it may not be always possible to choose $0.4n$ number of basepairs for every sequence in each iteration. In which case algorithm may go on sampling for infinite trials in iteration. This can be taken care by restricting number of consecutive failed trials (trials in which sampled basepair is not chosen since it is conflicting with some basepair already chosen in same iteration) to be some value. We set $maxFailedTrial = 0.25n$. Algorithm for sampling basepairs is shown as algorithm 1.

Random function in algorithm 1 is used for sampling multinomial random variables. *probDist* is array of $Prob(i, j), \forall (i, j) \in E$.

5.3 Time and Space complexities

In each iteration of this algorithm, there are $O(n)$ trials. For each trial, there is $O(\log n)$ effort involved (*Random* function of algorithm). This is because, basepairs are like multinomial random variables and after getting random value, algorithm performs binary search in *probDist* array to find basepair that has been picked up. Thus, each iteration

Algorithm 1 Algorithm for sampling basepairs.

Sample(probDist , maxIteration , maxFailedTrial)

Initialize $iteration = 1$, $maxScore = -1$

while $iteration \leq maxIteration$ **do**

  for $i = 1$ to $i = |E|$ **do**

    covered(i)=0

  end for

  tmpScore=0

  for $trial = 1$ to $trial < 0.4n$ **do**

    numFailedTrial=0

    while $0 \leq numFailedTrial$ and $numFailedTrial < maxFailedTrial$ **do**

      Choose $k = Random(1 : |E|, probDist)$

      $ii = k.i$

      $jj = k.j$

      if !$covered(ii)$ and !$covered(jj)$ **then**

        covered(ii)=TRUE

        covered(jj)=TRUE

        numFailedTrial=-1

        tmpScore = tmpScore + probDist(k)

      else

        numFailedTrial=numFailedTrial+1

      end if

    end while

  end for

  if $tmpScore > maxScore$ **then**

    bestSamples=covered

  end if

end while

return bestSamples

involves $O(n \log n)$ effort. We do fixed number of iterations, in which case time complexity is $O(n \log n)$ which is better compared to $O(n^3)$ time complexity of MWM algorithm. Even if number of iterations are made $O(n)$, overall time complexity with random sampling approach remains $O(n^2 \log n)$. On UNIX system with Pentium 4 processor with 1 GB RAM prediction of length around 1500 nucleotides (16SrRNA) takes about half minute.

5.4 Experimental Results and Evaluation

Experiments were done on real life datasets taken from Rfam and NCBI. Length of consensus sequences of multiple alignment of RNAs range from 100 to 1500 nucleotides. Table 5.2 lists details of sequences used in experiments. Quality of prediction is given in terms of sensitivity and specificity. Let RP be the number of base pairs in the reference structure, TP be the number of correctly predicted base pairs (true positives) and FP be the number of predicted base pairs that are not contained in the reference structure (false positives). Then *sensitivity* - *SS* and *specificity* - *SP* are calculated as $SS = \frac{100 \times TP}{RP}\%$ and $SP = \frac{100 \times TP}{TP+FP}\%$.

Table 5.3 shows SS and SP values over various sequences for different number of iterations. In this case, column number 2 to 5 with 100 , 200 , 300 and 400 iterations, all basepairs were considered. Whereas for last column marked as 30* , only those basepairs which account for first 80% of cumulative weights were considered. This approach rules out about 50% of basepairs, but still SS and SP do not drop that much and also it can be achieved in only 30 iterations.

We tested HxMatch on single sequences which did not have alignment information. For single sequence, that sequence itself can be considered as consensus sequence. This single sequence is given as alignment to HxMatch. The dataset has around 438 sequences with length of about 1500 nucleotides. Figure 5.3 shows frequency of different values for SS. Due to unavailability of alignment information, it does not perform quite good, which is expected. However, it is still better than dynamic programming approaches

RNA Family	N	Length	RP	PK
Tombusvirus	12	91	24	1
Enterovirus	12	103	38	1
SRP RNA	8	305	86	1
tmRNA	8	362	106	4
Rnase P RNA	8	404	124	2
16S rRNA	8	1542	478	2

Table 5.2: Sequences used for result. N is for number of sequences in alignment. RP is number of basepairs in reference structure. PK is number of pseudoknots in reference structure.

which cannot predict for such long sequences. Experiments for HxMatch with random sampling, instead of original HxMatch with MWM, were also done. In this case, all basepairs were considered. It still remains to be seen what happens when only basepairs with higher weights, which account for 80% of cumulative weights are used.

	Default		100		200		300		400		30*	
	SS	SP	SS	SP	SS	SP	SS	SP	SS	SP	SS	SP
Tombusvirus	37.5	39.1	37.5	47.4	54.2	56.5	54.2	56.5	50.0	52.2	54.2	68.4
Enterovirus	76.3	87.9	63.2	96.0	68.4	81.2	52.6	87.0	65.8	83.3	81.6	100.0
SRP RNA	91.9	86.8	76.7	84.6	80.2	88.5	73.3	85.1	72.1	88.6	93.0	88.9
tmRNA	80.2	90.4	57.5	91.0	50.9	77.1	53.8	80.3	64.2	91.9	75.5	89.9
Rnase P RNA	77.4	88.1	63.7	87.8	60.5	91.5	65.3	85.3	66.1	88.2	71.8	89.9
16S rRNA	77.6	85.7	48.5	86.2	51.7	88.2	46.4	83.5	46.2	85.7	50.6	88.6

Table 5.3: Results comparison with random sampling approach. In case of 100 , 200 , 300 , 400 iterations all basepairs with positive score were considered for sampling. Column with 30 iterations, shows results with only basepairs accounting initial 80% cumulative weight were considered.

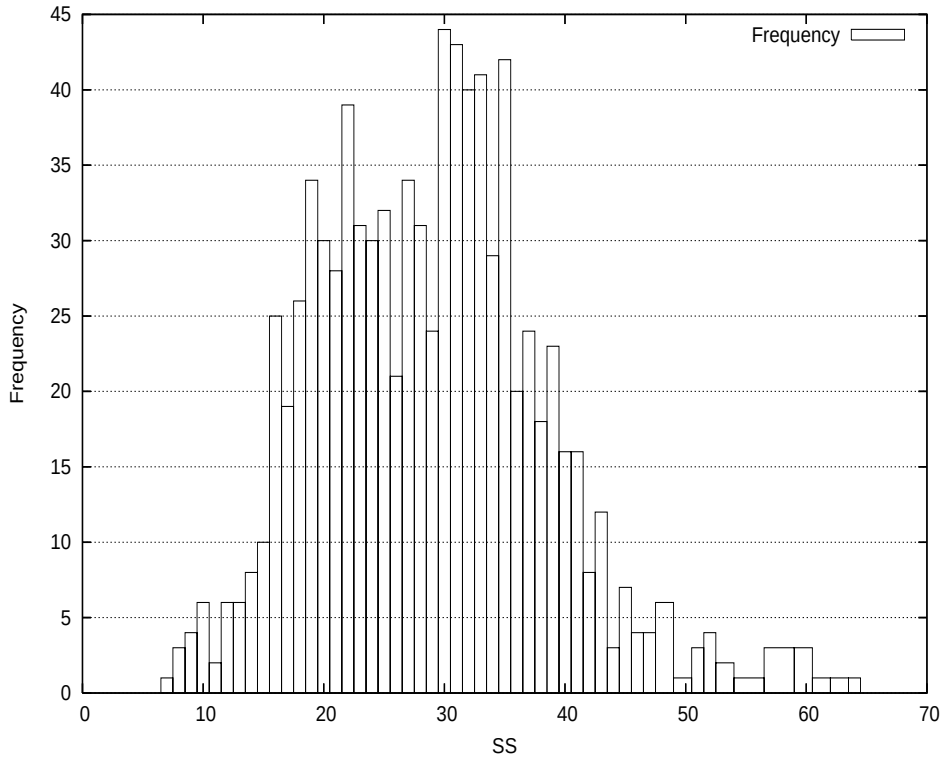


Figure 5.3: Frequency distribution of SS on single sequences of length around 1500 nucleotides.

Chapter 6

Discussion

We started this project with aim of predicting secondary structure for RNAs from reallife data with length above 700 nucleotides. Dynamic programming algorithms [7] [8] with time $O(n^4)$ and space $O(n^3)$ complexities are not suitable for such lengths. For 500 length sequence it can not produce results even after 2 hours. Recurrence for predicting simple pseudoknots results into $O(n^4)$ time and $O(n^3)$ space complexities. Existing algorithms try to predict simple pseudoknots over full RNA sequence, but simple pseudoknots generally occur over length of 100 nucleotides. So we propose two-stage dynamic programming algorithm, which first predicts regions of higher score with pseudoknots than non-pseudoknots. Only if region gives better score for pseudoknots than it goes for predicting actual structure.

Two-stage algorithm performs far better in terms of time, compared to Rivas' algorithm and Deogun's algorithm. For example, for sequence of length 121 nucleotides, it takes around 0.02 seconds, whereas Deogun's algorithm takes 8 minutes and Rivas' algorithm takes 4 hours and 30 minutes. Deogun et al [8] predicts 95% pseudoknots and with 78% correct or almost correct predictions. Two-stage algorithm predicts pseudoknot regions for 86% sequences, with 54% being correct or almost correct. Rivas' algorithm predicts only 50% pseudoknots.

As listed in Table 4.2, ignoring $G = U$ pairings and conflicts cause most of the incorrect predictions cases, around 35 out of 86. L matrix can be modified to consider

for $G = U$ pairings and consider some selected subsets of candidate stack pairs also to avoid conflicts. But in that case L matrix will have multi-valued entries and also sparseness will be reduced, affecting time of execution. Also, algorithm uses base-pairs score. Energy model score function can perform better compared to basepairs score. Only L matrix needs to be modified to hold energy values of stackpairs rather than base pairs score. Rest of the algorithm remains the same. Further speed-up can be achieved with hashing candidate stack pairs, L matrix entries.

First stage predicts only regions of pseudoknots and not the structure, over given RNA sequences. After getting pseudoknot regions information from first stage, actual structure can be predicted by applying Akutsu's algorithm on those regions to predict pseudoknot structure. However, overall accuracy of two-stage dynamic programming algorithm is largely dependent on performance of first stage.

Multiple alignment of sequences from same family of RNAs give useful information which helps in defining better score-matrix. Witwer et al [15] showed that using MWM to choose basepairs from score-matrix one can deal with large sequences. Based on our observations, we propose that instead of MWM, even with random sampling comparable accuracy can be achieved. Whereas dynamic programming approach just can not solve such problems, random sampling can solve it in less than one minute for sequence of length 1500.

There is still scope of improvement with random sampling. Like, we are allowing basepairs which account for initial 80% of cumulative weights. This is fixed for any sequence. Infact, instead of keeping it same for any sequence, it can be chosen to be some value depending upon some criterion. Figure 5.2 affirms this claim, only first 20% basepairs have comparable weights to rest of the basepairs. Also, only $5n$ or some $O(n)$ basepairs with maximum scores can be allowed. In future, this is the good direction to explore.

Another approach can be that random sampling prediction can act as starting point for dynamic programming algorithm. In this approach, basepairs selected randomly can be fed directly into dynamic programming algorithm and then it can try to predict more

basepairs. This makes sense also since FP (false positives) are not very high. This fact is reflected by SP measures (last column of Table 5.3).

References

[1] Waterman M.S., *Introduction to Computational Biology*, Chapman & Hall, London, 1995.

[2] Waterman M.S., Smith T.F., "Rapid Dynamic Programming Algorithms for RNA Secondary Structure", *Advances Applied Mathematics 7*, 455-464, 1986.

[3] Akiyama Y., Kanehisa. M., "NeuroFold: an RNA Secondary Structure Prediction System Using a Hopfield Neural Networks", *Proceedings of the Genome Informatics Workshop III*, Universal Academy Press, Tokyo, 199-202 (in Japanese).

[4] Brown M., Wilson C., "RNA Pseudoknots Modeling Using Intersections Of Stochastic Contex Free Grammars With Applications To Database Search", *L.Hunter, T.E.Klein (Eds.), Pacific Symposium On Bio-Computing 96*, World Scientific, Singapore, 109-125, 1996.

[5] Uemura Y., Hasegawa A., Kobayashi S., Yokomori T., "Tree Adjoining Grammars for RNA structure prediction", *Theoretical Computer Science 210*, 277-303, 1999.

[6] Rivas E., Eddy S.R., "A Dynamic Programming Algorithm For RNA Structure Prediction Including Pseudoknots", *Journal Of Molecular Biology 285*, 2053-2068, 1999.

[7] Akutsu T., "Dynamic Programming Algorithm For RNA Structure Prediction with Pseudoknots", *Discrete Applied Mathematics 104*, 45-62, 2000.

[8] Deogun J.S., Donis R., Komina O., Fangrui Ma, "RNA Secondary Structure Prediction With Simple Pseudoknots", 2^{nd} *Asia Pacific Bioinformatics Conference*, 239-246, 2004.

[9] Walter A., Turner D., Kim J., Lyttle M., Muller P., Mathews D., Zuker M., "Coaxial Stacking of Helixes Enhances Binding of Oligoribonucleotides and Improves Predictions of RNA Folding", *Proceedings of National Academy of Sciences*, 9218-9222, 1991.

[10] De Smit MH , Van Duin J., "Control of prokaryotic translation initiation by mRNA secondary structure", *Progress in Nucleic acid research in Molecular Biology 38* , 1-35, 1990.

[11] Mills DR, Priano C, Merz PA, and Binderow BD. Q ,"RNA bacteriophage: mapping cis-acting elements within an RNA genome" , *Journal Virol 64* , 3872-3881, 1990.

[12] Brannan CI, Dees EC, Ingram RS, and Tilghman SM. , "The product of the h19 gene may function as an RNA." *Molecular Cell Biology 10* , 28-36 , 1990.

[13] Brown CJ, Ballabio A, Rupert JL, Lafreniere RG, Grompe M,Tonlorenzi R, and Willard HF. , "A gene from the region of the human X inactivation center is expressed exclusively from the inactive X chromosome" , *Nature 349*, 38-44 ,1991.

[14] Zuker M., Mathews D.H., Turner D.H.. "Algorithms and Thermodynamics for RNA Secondary Structure Prediction: A Practical Guide", *RNA Biochemistry and Biotechnology*, J. Barciszewski & B.F.C. Clark, eds., NATO ASI Series, Kluwer Academic Publishers. Up- todate reproduction of the article is available at http://www.bioinfo.rpi.edu/zukerm/seqanal/.

[15] Christina Witwer and Ivo L. Hofacker, Peter F. Stadler. "Prediction of Consensus RNA Secondary Structures Including Pseudoknots", *IEEE/ACM Transactions on Computational Biology and Bioinformatics* , 66-77, 2004.

[16] J.E. Tabaska, R.B. Cary, H.N. Gabow, and G.D. Stormo, "An RNA Folding Method Capable of Identifying Pseudoknots and Base Triples", *Bioinformatics 14-8*, 691-699, 1998.

[17] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner, "Expanded Sequence Dependence of Thermodynamic Parameters Provides Robust Prediction of RNA Secondary Structure," *J. Molecular Biology 288*, 911-940, 1999.

[18] D.K. Chiu and T. Kolodziejczak, "Inferring Consensus Structure from Nucleic Acid Sequences," ĈABIOS 7, 347-352, 1991.

[19] I.L. Hofacker, M. Fekete, and P.F. Stadler, "Secondary Structure Prediction for Aligned RNA Sequences," *J. Molecular Biology 319,* 1059-1066, 2002.

[20] K. Doshi, J. Cannone, C. Cobaugh, and R. Gutell, "Evaluation of the Suitability of Free-Energy Minimization Using Nearest- Neighbor Energy Parameters for RNA Secondary Structure Prediction," *BMC Bioinformatics 5,* 105, 2004.

[21] D.A.M. Konings and R.R. Gutell, "A Comparison of Thermodynamic Foldings with Comparatively Derived Structures of 16S and 16S-Like rRNAs," *RNA 1,* 559-574, 1995.

[22] S.R. Morgan and P.G. Higgs, "Evidence for Kinetic Effects in the Folding of Large RNA Molecules," *J. Chemical Physics 105,* 7152-7157, 1996.

[23] O.V. Galzitskaya, "Geometrical Factor and Physical Reasons for Its Influence on the Kinetic and Thermodynamic Properties of RNA-Like Heteropolymers," *Folding and Design 2,* 192-201, 1997.

[24] O.V. Galzitskaya and A.V. Finkelstein, "Computer Simulation of Secondary Structure Folding of Random and Edited RNA Chains," *J. Chemical Physics 105,* 319-325, 1996.

[25] P.G. Higgs, "RNA secondary structure: physical and computational aspects," *Quarterly Rev. Biophysics 33,* no. 3, 199-253, 2000.

[26] I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhoeffer, M. Tacker, and P. Schuster, "Fast Folding and Comparison of RNA Secondary Structures," *Monatshefte fur Chemie 125,* 167-188, 1994.

[27] I.L. Hofacker, W. Fontana, P.F. Stadler, and P. Schuster, "Vienna RNA Package," *http://www.tbi.univie.ac.at/ivo/RNA/ Free Software,* 1994.

[28] M. Zuker and D. Sankoff, "RNA Secondary Structures and Their Prediction," *Bull. Math. Biology 46,* 591-621, 1984.

[29] C. Zwieb, I. Wower, and J. Wower, "Comparative Sequence Analysis of tmRNA," *Nucleic Acids Research 27,* no. 10, 2063-2071, 1999.

[30] H.N. Gabow, "Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs," *PhD thesis, Stanford Univ.,* 1973.

[31] E. Rothberg, "Solver for the Maximum Weight Matching problem," *ftp://dimacs.rutgers.edu/pub/netflow/matching/ weighted/solver-1,* 1985.

[32] W. Fontana, D.A. M. Konings, P.F. Stadler, and P. Schuster, "Statistics of RNA Secondary Structures," *Biopolymers 33,* 1389-1404, 1993.

[33] I.L. Hofacker, M. Fekete, C. Flamm, M.A. Huynen, S. Rauscher, P.E. Stolorz, and P.F. Stadler, "Automatic Detection of Conserved RNA Structure Elements in Complete RNA Virus Genomes" *Nucleic Acids Research 26,* 3825-3836, 1998.

[34] J. Ruan, G.D. Stormo, and W. Zhang, "An Iterated Loop Matching Approach to the Prediction of RNA Secondary Structures with Pseudoknots," *Bioinformatics 20,* 58-66, 2004.