Summer 8-31-2005

RNA structure analysis : algorithms and applications

Jianghui Liu

New Jersey Institute of Technology

ABSTRACT

RNA STRUCTURE ANALYSIS: ALGORITHMS AND APPLICATIONS

by
Jianghui Liu

In this doctoral thesis, efficient algorithms for aligning RNA secondary structures and mining unknown RNA motifs are presented. As the major contribution, a structure alignment algorithm, which combines both primary and secondary structure information, can find the optimal alignment between two given structures where one of them could be either a pattern structure of a known motif or a real query structure and the other be a subject structure.

Motivated by widely used algorithms for RNA folding, the proposed algorithm decomposes an RNA secondary structure into a set of atomic structural components that can be further organized in a tree model to capture the structural particularities. The novel structure alignment algorithm is implemented using dynamic programming techniques coupled by position-independent scoring matrices. The algorithm can find the optimal global and local alignments between two RNA secondary structures at quadratic time complexity. When applied to searching a structure database, the algorithm can find similar RNA substructures and therefore can be used to identify functional RNA motifs. Extension of the algorithm has also been accomplished to deal with position-dependent scoring matrix in the purpose of aligning multiple structures.

All algorithms have been implemented in a package under the name *RSmatch* and applied to searching mRNA UTR structure database and mining RNA motifs. The experimental results showed high efficiency and effectiveness of the proposed techniques.

RNA STRUCTURE ANALYSIS: ALGORITHMS AND APPLICATIONS

by
Jiangui Liu

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

August 2005

APPROVAL PAGE

RNA STRUCTURE ANALYSIS: ALGORITHMS AND APPLICATIONS

Jianghui Liu

Dr. Jason T.L. Wang, Dissertation Advisor — Date
Professor of Computer Science, New Jersey Institute of Technology

Dr. Bin Tian, Dissertation Co-Advisor — Date
Assistant Professor of Bioinformatics & Molecular Biology, University of Medicine and Dentistry of New Jersey

Dr. James McHugh, Committee Member — Date
Professor of Computer Science, New Jersey Institute of Technology

Dr. David Nassimi, Committee Member — Date
Associate Professor of Computer Science, New Jersey Institute of Technology

Dr. Marc Q. Ma, Committee Member — Date
Assistant Professor of Computer Science, New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author: Jianghui Liu

Degree: Doctor of Philosophy

Date: August 2005

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2005
- Master of Engineer in Computer Engineering,
  Beijing University of Posts and Telecommunications, Beijing, P.R.China, 1999
- Bachelor of Science in Computer Engineering,
  Nanjing Institute of Posts and Telecommunications, Nanjing, P.R.China, 1996

Major: Computer Science

Presentations and Publications:

Jianghui Liu, Jason T.L. Wang and Bin Tian, "Mining conserved RNA stem-loops in human and mouse UTRs" *submitted*

Bin Tian and Jianghui Liu, "Alternative mRNA polyadenylation affects detection of gene expression by Affymetrix GeneChips", *submitted*

Jianghui Liu, Jason T.L. Wang, Jun Hu and Bin Tian, "A method for aligning RNA secondary structures and its application to RNA motif detection," *BMC Bioinformatics*, vol. 6, 89:1–31, 2005.

Jianghui Liu, Jason T.L. Wang, Wynne Hsu and Katherine G. Herbert, "XML Clustering by Principle Component Analysis," In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pp. 658–662, Boca Raton, FL, 2004.

Katherine G. Herbert, Jason T.L. Wang and Jianghui Lui, "Information Retrieval and Data Mining," *The Computer Science and Engineering Handbook*, Second Edition, CRC Press, 2004.

Jianghui Liu, Jun Hu, Jason T.L. Wang and Bin Tian, "Mining RNA Structural Elements in UTRs," Poster on the *Identification of Functional Elements in Mammalian Genomes*, Cold Spring Harbor Laboratory, NY, Nov. 11–13, 2004.

This dissertation is dedicated to my family for the love I enjoy in my life.

ACKNOWLEDGMENT

I would like to warmly thank my advisor, Dr. Jason T.L. Wang, for his consistent support and encouragement through my five-year PhD study. My thanks also extend to Dr. Bin Tian from University of Medicine and Dentistry of New Jersey for giving me generous support for my last year research work and giving me opportunities to enjoy bioinformatics.

I want to thank the committee members, Dr. Bin Tian, Dr. James McHugh, Dr. David Nassimi and Dr. Marc Q. Ma, for their precious time and in-depth reviews to help me accomplish this wonderful project.

I want to thank all my colleagues and collaborators for their friendship in the past several years and greatly appreciate the in-depth criticism and advice for my research and daily life from the following persons: Dr. Haibo Zhang, Ju Hu, Zhenhua Pan, Shouxian Chen, Wugang Xu, Spencer Zhou, Yi Meng, Dongrong Wen, Junhan Wang and Dr. Katherine Herbert.

I would also like to thank my wife for her enduring patience and trust in me. My families in China have been a great source of comfort whenever I was in trouble.

Finally, I thank everybody whoever helped me to finish this great project.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Consider the following problem: given two RNA secondary structures, what is the maximum common substructures? Since formal definition of RNA *secondary structure* will be given in the following chapters, at this time it can simply assume that an RNA secondary structure is composed of two types of components: single bases and bonded base-pairs which are somehow assembled to form the ultimate structure. This problem, like finding the longest common subsequence between two strings [1], could have been of great importance to several areas of computational biology [2].

Coupled with the availability of genomic information of more and more species, especially homo sapiens, efficient computing technologies are valuable and desperately demanded by biologists for bio-data analysis. In the past years, significant discoveries had been achieved at sequential or primary structure level [3]. The milestone work on general purpose algorithms for sequence analysis was accomplished by Hirschberg [4], also known in the computational biology community as the Myers/Miller algorithm [5].

Sequence oriented analysis tools, like BLAST [6,7], FASTA [8], CLUSTALW [9], have achieved great popularity in protein and DNA research communities. It has become a routine task for researchers to use these tools in purpose of finding homologs and searching databases. These tools are sequence-based in that only sequence or primary structure information is used in the analysis, while no or very tiny secondary or higher level structural information being considered.

However, for research work related to RNAs, sequence-based methods do not work well in most cases. One of the most important reasons is that nucleotide bases do not carry as much functional/structural information as amino acid residues do.

RNA molecule's structure, hence its function, is determined jointly by both sequence composition and distant interactions among bases [10]. A typical example is RNA motif detection: unlike protein motif searching which can be accomplished through the development of sophisticated amino acid substitution matrices, computationally detecting or discovering RNA motifs is still at a primitive stage without broadly accepted methodologies in literatures.

For the purpose of discovering new RNA motifs, one promising approach would be to design an efficient algorithm that is able to align RNA secondary structures at the structurally conserved regions that could be treated as putative motifs for further experimental verification.

In this dissertation, efficient algorithms are presented to align RNA secondary structures and explore RNA motifs. As the major contribution, a structure alignment algorithm called RSmatch, which combines both primary and secondary structure information, can find the optimal alignment between two given structures where one of them could be either a pattern structure of a known motif or a real query structure and the other be a subject structure.

Motivated by widely used algorithms for RNA folding, the RSmatch algorithm decomposes an RNA secondary structure into a set of atomic structural components that can be further organized in a tree model to capture the structural particularities. The novel structure alignment algorithm is implemented using dynamic programming techniques coupled by position-independent scoring matrices. RSmatch can find the optimal global and local alignments between two RNA secondary structures at quadratic time complexity. When applied to searching a structure database, the algorithm can find similar RNA substructures and therefore can be used to identify functional RNA motifs. Extension of the algorithm has also been accomplished to deal with position-dependent scoring matrix for the purpose of conducting multi-structure alignment.

1.1 Biology Fundamentals

Many essential biological roles are found to be assumed by RNAs (RiboNucleic acid) molecules. With respect to different functions, RNAs are grouped into many families. Important families include transfer RNA (tRNA), ribosomal RNA (rRNA), small nuclear RNA in the spliceosome (snRNA), messenger RNA (mRNA), and various classes of introns. On the other hand, it is a known fact that majority of essential macromolecules are polymers of smaller constitutional components. For instance, RNA is a polymer of four types of nucleotides: **A**denine, **U**racil, **C**ytosine, **G**uanine; for protein, there exist 20 types of amino acid residues which are building blocks to form various types of proteins. These building blocks concatenate with each other to form a strand which finally folds back to itself to form complex three-dimensional structure. The strand form could be thought as the initial state of an RNA molecule. However, for most non-coding RNAs, the proper function is only enabled when the strand folds back to itself to form particular spatial conformation. Unfortunately, the complexity of exploring 3D spatial structures directly is prohibitive. Since the sequence composition gives some information of the final folded structure, much of the related research work had been done at sequence level.

At the sequence level, one important research field is to find out how similar two sequences could be [11,12]. The most popular method of similarity measure is through edit distance borrowed from string comparison [13,14]. Given two sequences $\mathbf{x}$ and $\mathbf{y}$, it tries to find an optimal series of transcriptions to transform $\mathbf{x}$ to $\mathbf{y}$ through three types of edit operations: substitute, insert, delete:

Substitute: one character in $\mathbf{x}$ is replaced by a corresponding character in $\mathbf{y}$.

Insert: One character in $\mathbf{y}$ is inserted into $\mathbf{x}$, causing $\mathbf{x}$ "grows" by one character.

Delete: One character in $\mathbf{x}$ is deleted, causing $\mathbf{x}$ "shrinks" by one character

The edit distance calculation can be illustrated in a two-dimensional matrix as shown in Figure 1.1 where the row is corresponding to characters in sequence **x** and the column to those of **y**. Each cell of the matrix represents an edit distance between the two prefix subsequences terminated by the cell's row- and column- indices respectively. Using $\mathcal{C}$ to denote the matrix, then the first row $\mathcal{C}[0, i]$ and first column $\mathcal{C}[i, 0]$ are initialized respectively by continuous integers starting from 0 (refer to Figure 1.1) . Using $\delta(.,.)$ to represent a generalization of the Kronecker delta function such that it has value 1 if the two character arguments are identical and 0 otherwise. The rest cells of the matrix are calculated as:

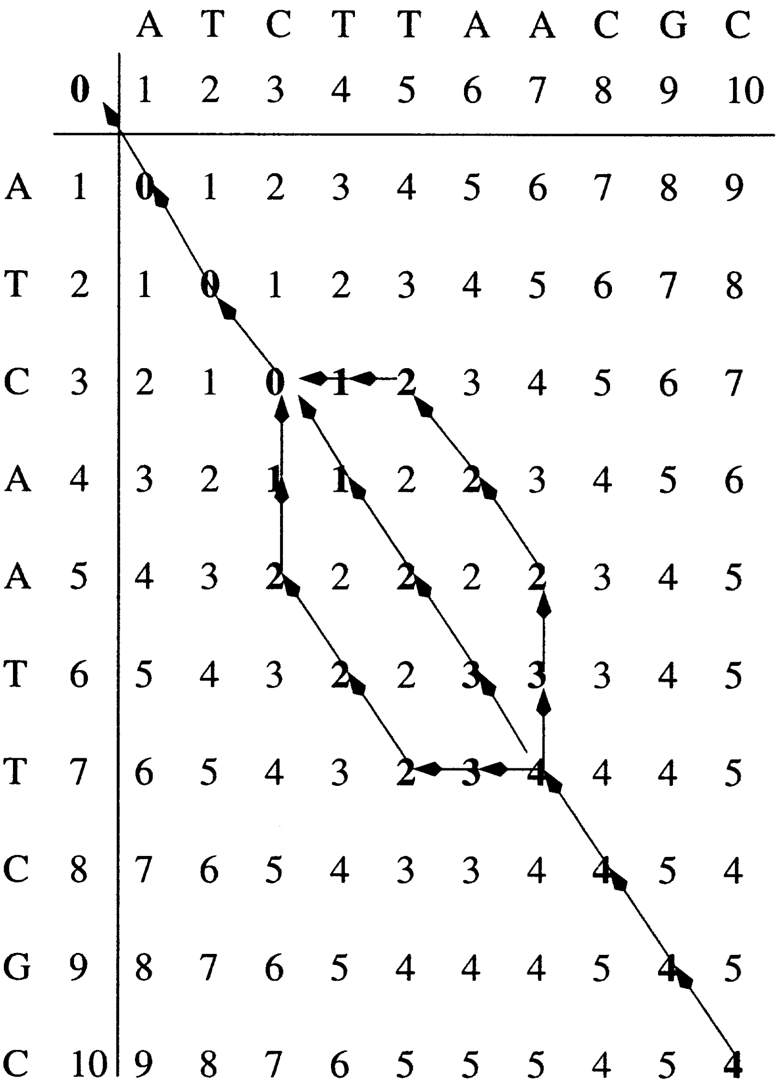
$$\mathcal{C}[i,j] = \min \left[\underbrace{\mathcal{C}[i-1,j]+1}_{delete}, \underbrace{\mathcal{C}[i,j-1]+1}_{insert}, \underbrace{\mathcal{C}[i-1,j-1]+1-\delta(\mathbf{x}[i],\mathbf{y}[j])}_{nochange/substitue} \right]$$

Beyond the similarity among sequences, a further and more interesting step to biologists is to find alignment between two sequences or among several sequences (multi-alignment). Alignment among sequences could provide biological inference for the construction of phylogenies, structure/function prediction and homolog searching.

Alignment of two sequences can be achieved by similar algorithm as that of computing edit distances, also known in the computational biology community as the Myers/Miller algorithm [5]. The algorithm is derived from considering the following three ways an alignment column could adopt:

- $\mathbf{x}[i]$ aligned with nothing, which is equivalent to "delete" operation;
- $\mathbf{y}[j]$ aligned with nothing, which is equivalent to "insert" operation;
- $\mathbf{x}[i]$ aligned with $\mathbf{y}[j]$, which is equivalent to "substitute" operation;

Furthermore, with each alignment situation an alignment cost shall be associated properly. Then the calculation follows similarly as that of edit distance. When the



		A	T	C	T	T	A	A	C	G	C
	0	1	2	3	4	5	6	7	8	9	10
A	1	0	1	2	3	4	5	6	7	8	9
T	2	1	0	1	2	3	4	5	6	7	8
C	3	2	1	0	1	2	3	4	5	6	7
A	4	3	2	1	1	2	2	3	4	5	6
A	5	4	3	2	2	2	2	2	3	4	5
T	6	5	4	3	2	2	3	3	3	4	5
T	7	6	5	4	3	2	3	4	4	4	5
C	8	7	6	5	4	3	3	4	4	5	4
G	9	8	7	6	5	4	4	4	5	4	5
C	10	9	8	7	6	5	5	5	4	5	4

Figure 1.1 An example of edit distance calculation. The computation fills the matrix row by row until the lower right corner cell is reached which gives the full edit distance between two sequences. The edit distance between "ATCAATTCGC" and "ATCTTAACGC" is thus 4. The transcription series can be recovered by tracing back the calculation path until the top left corner cell is reached. As shown, several paths are equivalent in the sense of giving the same edit distance value.

optimal score is reached at the lower right corner of the table, the trace-back is needed to reveal the optimal alignment. It is very well possible for several optimal alignments having the same optimal score. In this case, if without interference of expertise, all of the alignments shall be treated equally.

If the costs to alignment situations are assigned by the following function:

$$f(\mathbf{x}[i], \mathbf{y}[j]) = \begin{cases} 1 & \text{if } \mathbf{x}[i] = \mathbf{y}[j] \\ -1 & \text{if } \mathbf{x}[i] \neq \mathbf{y}[j] \\ -1 & \text{if either } \mathbf{x}[i] \text{ or } \mathbf{y}[j] \text{ is empty} \end{cases}$$

The alignment calculation of the same two sequences as Figure 1.1 is illustrated in Figure 1.2. Two optimal alignments are:

ATC–TTAACGC

ATCAATT–CGC

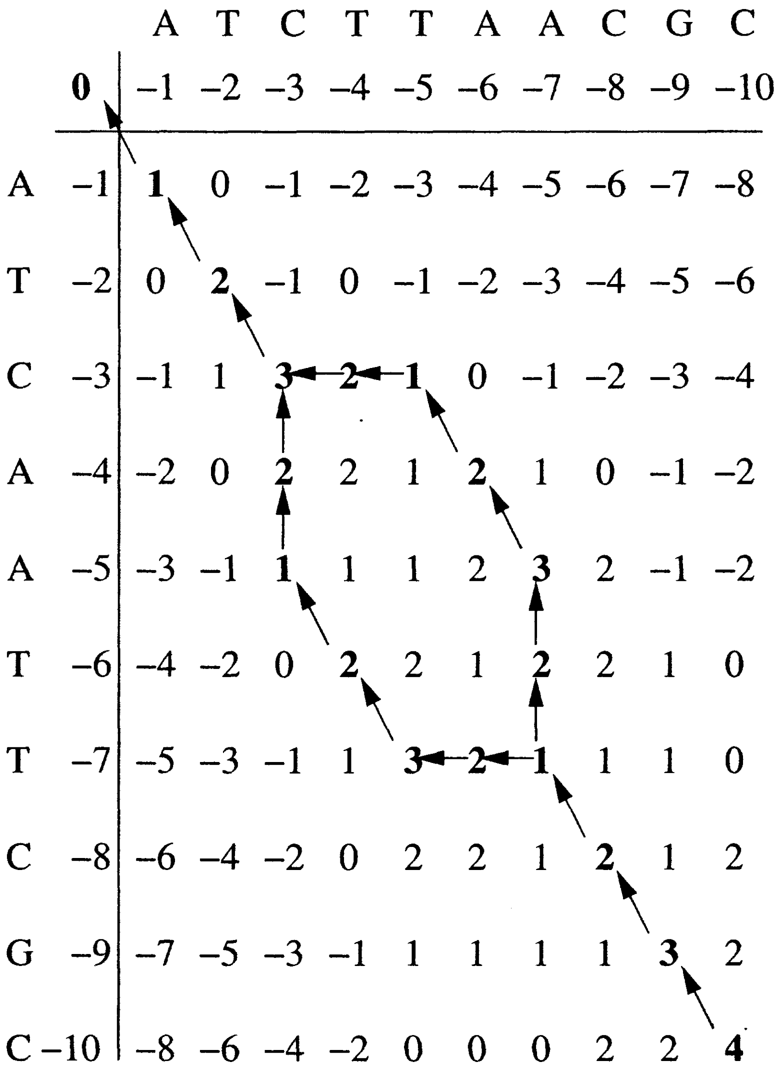
and

ATCTTAA–CGC

ATC–AATTCGC

The above alignments are called *global* alignment because all letters of both sequences are involved in the alignment. *Local alignment* only considers the alignment between two subsequences. However, the algorithm is very similar to the *global* version with three differences: first, the initialization will set all cells in the first row and first column to zeros; second, to decide the value for current cell, if the best score of aligning the corresponding two subsequences is negative, the cell's score will be set to 0; third, the tracing process will start from the cell with the maximum score instead of the lower right corner cell and will end at the cell having negative score.

Mayers/Miller algorithm can be improved by heuristic methods, as done in most popular sequence analysis tools, i.e. BLAST [7], FASTA [8], CLUSTALW [9]. Practically, the heuristic approaches run much faster but still produce the same, or



		A	T	C	T	T	A	A	C	G	C
	0	−1	−2	−3	−4	−5	−6	−7	−8	−9	−10
A	−1	**1**	0	−1	−2	−3	−4	−5	−6	−7	−8
T	−2	0	**2**	−1	0	−1	−2	−3	−4	−5	−6
C	−3	−1	1	**3**	**2**	**1**	0	−1	−2	−3	−4
A	−4	−2	0	**2**	2	1	**2**	1	0	−1	−2
A	−5	−3	−1	**1**	1	1	2	**3**	2	−1	−2
T	−6	−4	−2	0	**2**	2	1	**2**	2	1	0
T	−7	−5	−3	−1	1	**3**	**2**	**1**	1	1	0
C	−8	−6	−4	−2	0	2	2	1	**2**	1	2
G	−9	−7	−5	−3	−1	1	1	1	1	**3**	2
C	−10	−8	−6	−4	−2	0	0	0	2	2	**4**

Figure 1.2 An example *global* sequence alignment. The computation fills the matrix row by row until the lower right corner cell is reached which presents the final alignment score. The *global* alignment score between "ATCAATTCGC" and "ATCTTAACGC" is thus 4. The detailed character-by-character alignment can be formed by tracing back calculation path until the top left corner cell is reached. The optimal alignment is not unique. Two optimal alignments are highlighted by the tracing arrows shown in the diagram.

almost the same results as that of Mayers/Miller algorithm. Currently, the primary goal of most of the proposed general-purpose alignment algorithms is to find homologs or consensus from bio-sequence databases.

1.2 Previous Works on RNA Structures

Well known by biologist, functions of macromolecules largely depend on secondary and/or tertiary structures. This observation is clearly manifested in most enzyme catalytic and virus invading mechanisms. It is biologically reasonable that homologs or consensus represented in the form of secondary and/or tertiary structure would be more important and informative than those in the form of primary structure [15]. However, direct structural analysis is considered difficult, if not impossible. One important reason is the lack of efficient algorithm to analyze large macromolecules directly at structure level.

Furthermore, recent years have witnessed the great achievement in constructing gene regulation networks, in which the post-transcriptional regulatory signals play important roles. In mRNA molecule, myriad of these signals, known as RNA motifs, have been detected in the UTR regions. These regulatory motifs play a variety of roles for post-transcriptional gene regulation which involve modulation of RNA localization, translation and stability [16, 17]. The regulation functions are mainly accomplished by interactions between the motifs and relevant proteins via binding machinery. RNA motifs are distinguished from DNA-mediated regulation signals, whose biological activity is essentially mediated by their primary structure. The activity of RNA motifs heavily depends on a combination of primary and secondary structure [18].

Great efforts at sequence level have been made in the detection of RNA motifs [10, 19–25]. Tools that align biosequences (DNA, RNA, protein), such as FASTA and BLAST, are valuable in identifying homologous regions, which can lead to the discovery of functional units, such as protein domains, DNA cis-elements, etc. [6,8].

Other first-order sequence analysis tools are following the direction of probabilistic model. Hidden Markov Models (HMMs) are proposed and applied to model protein families and domains successfully [26]. These models treat each position along the sequence having independent distributions. Strictly speaking, HMM models are only able to deal with first-order correlations between two consecutive positions. This is reasonable for protein sequence analysis and practically shows its feasibility.

The success of HMM models is more evident in the study of DNAs and proteins than of RNAs. This is mainly because the sequence similarity for DNAs sequences or protein sequences can usually faithfully reflect their functional relationship, whereas additional structure information is needed to study the functional conservation for RNAs. Therefore, it is necessary to take into account both structural and sequential information in analyzing RNA sequences.

Alignment of RNA secondary structures is important in studying functional RNA motifs. In recent years, much progress has been made in RNA motif finding and structure alignment. However, existing tools either require a large number of prealigned structures or suffer from high time complexities. This makes it difficult for the tools to process RNAs whose prealigned structures are unavailable or process very large RNA structure databases.

An efficient tool called RSmatch is presented in this thesis for aligning RNA secondary structures and for motif detection. Motivated by widely used algorithms for RNA folding, this study decomposes an RNA secondary structure into a set of atomic structure components that are further organized by a tree model to capture the structural particularities. RSmatch can find the optimal global and local alignment between two RNA secondary structures using two scoring matrices, one for single-stranded regions and the other for double-stranded regions. The time complexity of

RSmatch is $\mathcal{O}(mn)$ where m is the size of the query structure and n that of the subject structure. When applied to searching a structure database, RSmatch can find similar RNA substructures, and is capable of conducting multiple structure alignment and iterative database search. Therefore it can be used to identify functional RNA motifs. The accuracy of RSmatch is tested by experiments using a number of known RNA structures, including simple stem loops and complex structures containing junctions.

With respect to computing efficiency and accuracy, RSmatch compares favorably with other tools for RNA structure alignment and motif detection. This tool shall be useful to researchers interested in comparing RNA structures obtained from wet lab experiments or RNA folding programs, particularly when the size of the structure dataset is large.

CHAPTER 2

FUNDAMENTAL OF RNA SECONDARY STRUCTURE

In nature, RNA molecule exists in the form of 3D structure by folding back to itself. Unfortunately, the 3D conformation is too untractable for analysis purpose, especially in large-scale. At the utmost fundamental level, RNA is a string of nucleotides concatenated with each other. Instead of targeting at RNA sequences, researcher are more interested in the folded RNAs whose structure complex is moderate, *i.e.* secondary structure. This chapter will focus on the fundamentals of RNA secondary structure. The emphasis will be put on the circle-based structure decomposition model and how the model facilitates the algorithm design in the purpose of structure analysis.

2.1 RNA Structure Elements

RNA is usually represented as a sequence of nucleotide bases. In living cells, non-coding RNA strand will folds back to itself instead of staying as a plain string. The result of the folding process is a particular conformation which ultimately enables the RNA's proper function. Typical example is tRNA's cloverleaf conformation which is required to "clamp" amino acid residues shifting around within cytoplasm.

In contrast to protein, folded RNAs have relatively simpler interactions among bases, and hence simpler notations for the folded structure. When RNA strand folds back to itself, majority of the interactions between nucleotide bases are canonical Watson-Crick bonds which are formed between nucleotides Adenine and Uracil or between Cytosine and Guanine. Few non-canonical bonds, *i.e.* Uracil versus Guanine bond, could also be found. In most cases, non-coding RNA only functions properly until some particular conformation is formed after the folding machinery.

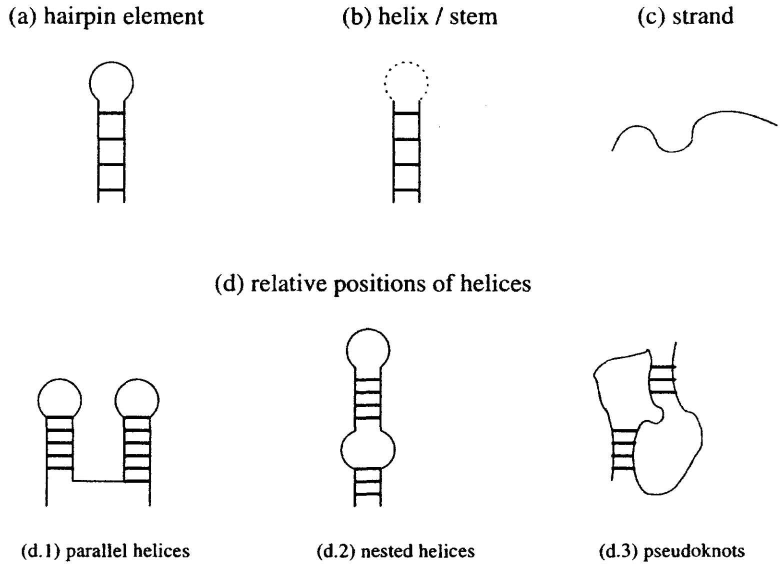


Figure 2.1 Basic structure elements of RNA secondary structure.

A very interesting thing with RNA folding is that an RNA sequence can fold into several conformations adapted to changed physical circumstances. Thus, one RNA may assume several functions under different *in vivo* environments. This points out that the research work about RNA functions can not be confined at sequence level. Both sequence and structure information are indispensable to delineate the panorama of RNA functions.

RNA crystallogram reveals that RNA structures can be described by a set of basic structure elements. Furthermore, these structure elements are depicted by a set of base interactions where majority of them involve only two bases. For an RNA structure notation, if no interaction concerns more than two bases, the description is at the level of secondary structure and the structure is called secondary structure [10]. In Fig.2.1, some basic structure elements are illustrated.

This dissertation accommodates all the secondary structure elements except pseudoknot element. Pseudoknot is not compatible with the loop decomposition process commonly used in RNA structure prediction tools [27,28]. It is excluded by *mfold* package also since it ruins the underlying structure prediction rules by breaking down the loop decomposition process. For the similar reason, pseudoknots are also excluded in this dissertation.

If pseudoknots are not allowed, the definition of secondary structure can be formalized straightforward [29]. An RNA secondary structure **R**, with its sequence as $\mathbf{R} = r_1, r_2, ..., r_n$, has two collections of structural components. One is a collection of base pairs denoted as (i, j), where for each pair there exists a bond connecting bases r_i and r_j. The other is a collection of single bases to which no bond is related. Formally, the following constraints are imposed:

- Any two base pairs are either identical or don't have base in common. This condition specifies that any nucleotide base can involve in up to one bond/pair;
- If there is a U-turn, biologically called hairpin loop, there must be at least 3 bases on the loop;
- Pseudoknots are precluded. That means, for any pair $p_i : (i_1, i_2)$ and $p_j : (j_1, j_2)$, the derived two sequence intervals $[i_1, i_2]$ and $[j_1, j_2]$ are either non-overlapping with each other or one is completely contained within the other. This condition guarantees that the RNA secondary structure can be draw on a plane without any bond crossing each other. It also means that any RNA secondary structure should be able to be depicted by the notation of nested parenthesis format as done in [30]. The nested parenthesis format is also adopted as the output format of *RSmatch*.

Based on the above formalization, different ways of representation are proposed as shown in Fig.2.2:

Bonds representation : This is the most intuitive representation of a secondary structure shown in Fig.2.2(a). Bonds between bases are highlighted and the relative relationships among structural components are obvious;

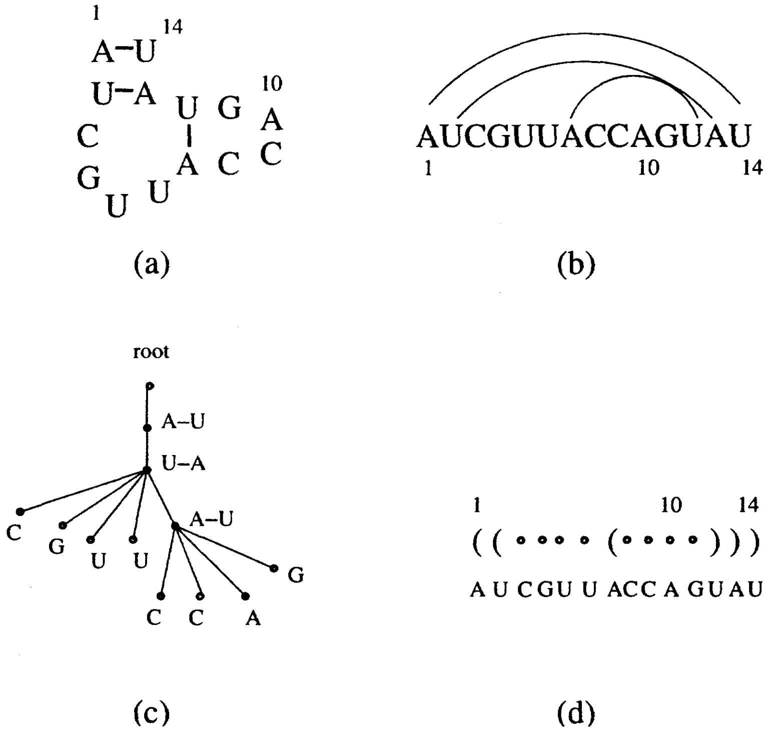


Figure 2.2 Several notations of RNA secondary structure.

Nested parenthesis representation : It is initially proposed by Hofacker *et al.* [31]. It is a compact structure representation by creating a string of the same length as the "unfolded" RNA sequence. The created string consists of parentheses and dots by replacing each base pair (i, j) with "(" and ")" in the ith and jth positions respectively, and replacing those single bases with ".". An example is given in Fig.2.2(d).

Tree representation : Various approaches of representing secondary structures as trees are proposed [31–33]. They differ from each other in that some representations compress substructures into single labeled nodes. Fig.2.2(c) shows the corresponding tree representation for the same structure by the nested parenthesis notation. The tree is ordered in that the order among siblings is important to distinguish each other. The tree root does not represent any part of the structure. Base pairs are corresponding to internal nodes and single bases are represented by leaves.

Arc representation : In this representation, the secondary structure is a set of non-crossing arcs connecting bonded bases. (see Fig.2.2(b)). These arcs are either parallel with each other or nested with each other [34].

This dissertation shows particular interest in the *Bonds Representation.* In fact, based on the *bonds representation*, an extended loop decomposition process, called *circle decomposition*, was proposed to decompose a secondary structure into a set of circles, which were further organized into a tree-like hierarchy.

CHAPTER 3

PREVIOUS WORK ON RNA SECONDARY STRUCTURE ANALYSIS

Alignment of RNA secondary structures is important in studying functional RNA motifs. In recent years, much progress has been made in RNA motif finding and structure alignment. However, existing tools either require a large number of prealigned structures or suffer from high time complexities. This makes it difficult for the tools to process RNAs whose prealigned structures are unavailable or process very large RNA structure databases.

Ribonucleic acid (RNA) plays various roles in the cell. Many functions of RNA are attributable to their structural particularities (herein called RNA motifs). RNA motifs have been extensively studied for noncoding RNAs (ncRNAs), such as transfer RNA (tRNA), ribosomal RNA (rRNA), small nuclear RNA (snRNA), small nucleolar RNA (snoRNA), etc. [35]. More recently, small interfering RNA (siRNA) and microRNA (miRNA) have been under intensive studies [36]. Less well characterized are the structures in the un-translated regions (UTRs) of messenger RNAs (mRNAs) [37]. However, biochemical and genetic studies have demonstrated a myriad of functions associated with the UTRs in mRNA metabolism, including RNA translocation, translation, and RNA stability [16, 38, 39].

RNA structure determination via biochemical experiments is laborious and costly. Predictive approaches are valuable in providing guide information for wet lab experiments. RNA structure prediction is usually based on thermodynamics of RNA folding or phylogenetic conservation of base-paired regions. The former uses thermodynamic properties of various RNA local structures, such as base pair stacking, hairpin loop, and bulge, to derive thermodynamically favourable secondary structures. A dynamic programming algorithm is used to find optimal or suboptimal structures. The most well-known tools belonging to this group are MFOLD [28] and RNAFold in the Vienna RNA package [39, 40]. Similar tools have been developed in recent years to predict higher order structures, such as pseudoknots [41]. On the other hand, RNA structure prediction using phylogenetic information infers RNA structures based on covariation of base-paired nucleotides [42–45]. It is generally believed that methods using phylogenetic information are more accurate. However, their performance critically depends on the high quality alignment of a large number of structurally related sequences.

Tools that align biosequences (DNA, RNA, protein), such as FASTA and BLAST, are valuable in identifying homologous regions, which can lead to the discovery of functional units, such as protein domains, DNA cis elements, etc. [6, 8]. However, their success is more evident in the study of DNAs and proteins than of RNAs. This is mainly because the sequence similarity among DNAs and proteins can usually faithfully reflect their functional relationship, whereas additional structure information is needed to study the functional conservation among RNAs. Therefore, it is necessary to take into account both structural and sequential information in comparing RNA sequences.

Several tools are available that carry out RNA alignment and folding at the same time (Table 3.1). The pioneer work by Sankoff [46] involves simultaneous folding and aligning of two RNA sequences, and has huge time and space complexity (Table 3.1). FOLDALIGN [47] improves the Sankoff's method by (1) scoring the structure solely based on the number of base pairs, instead of the stacking energies; and (2) disallowing branch structures (junctions). Dynalign [48] reduces the time complexity by restricting the maximum distance allowed between aligned nucleotides in two structures. By taking into account local similarity, stem energy and covariations, Perriquet et al. [49] proposed CARNAC for pairwise folding of RNA sequences. Ji et al. [50] applied a graph-theoretical approach, called comRNA, to detect the common

Table 3.1 Performance comparison of RNA secondary structure analysis tools

tool name	running time requirement	space requirement
Sankoff[1]	$\mathcal{O}(n^6)$	$\mathcal{O}(n^4)$
FOLDALIGN[2]	$\mathcal{O}(n^4)$	$\mathcal{O}(n^4)$
RAGA[3]	$\mathcal{O}(m^2n^3)$	$\mathcal{O}(m^2n^3)$
rna_align[4]	$min\{\mathcal{O}(mn^3), \mathcal{O}(m^3n)\}$	$\mathcal{O}(mn^2)$
Dyalign[5]	$\mathcal{O}(m^3n^3)$	$\mathcal{O}(m^2n^2)$
stemloc[6]	$\mathcal{O}(lm)$	N/A
Rsearch[7]	$\mathcal{O}(m^3n)$	$\mathcal{O}(m^3)$
RNAforester[8]	$\mathcal{O}(\|F_1\|\|F_2\|deg(F_1)deg(F_2)(deg(F_1)$ $+deg(F_2)))$	$\mathcal{O}(\|F_1\|\|F_2\|deg(F_1)deg(F_2))$
CARNAC[9]	$\mathcal{O}(n^6), \mathcal{O}(n^2)$	$\mathcal{O}(n^4), \mathcal{O}(n^2)$
comRNA[10]	$\mathcal{O}(m^n)$	N/A

[1] n is the sequence length;
[2] n is the average length of a given set of RNAs;
[3] m and n are the lengths of the two given sequences;
[4] m and n are the two sequence lengths;
[5] m is the maximum distance allowed to match two nucleotides and n is the length of the shorter sequence;
[6] l and m are the two RNA sequence lengths; only valid under extreme cases;
[7] m is the query length and n is the subject sequence length;
[8] $|F_i|$ is the number of nodes in forest F_i and $deg(F_i)$ is the degree of F_i;
[9] n is the sequence length, theoretical complexity is listed before practical complexity;
[10] m is the maximum number of stems examined and n is the number of total sequences under analysis.

RNA secondary structure motifs from a group of functionally or evolutionally related RNA sequences. One noticeable advantage of comRNA is its capability to detect pseudoknot structures. In addition, algorithms using derivative-free optimization techniques, such as genetic algorithms and simulated annealing, have been proposed to increase the accuracy in structure-based RNA alignment [51–53]. For example, Notredame et al. [51] presented RAGA to conduct alignment of two homologous RNA sequences when the secondary structure of one of them was known. As shown in Table 3.1, most of these methods suffer from high time complexities, making the structure-based RNA alignment tools much less efficient than sequence-based alignment tools.

Tools that search for optimal alignment for given structures include RNAdistance [2], rna_align [54], and RNAforester [55]. RNAdistance uses a tree-based model to coarsely represent RNA secondary structures, and compared RNA structures based on edit distance. In a similar vein, rna_align [54] models RNA secondary structures by nested and/or crossing arcs that connect bonded nucleotides. With the crossing arcs, rna_align is able to align two RNA secondary structures, one of which could contain pseudoknots. RNAforester extends the tree model to forest model, which significantly improves both time and space complexities (Table 3.1). In addition, methods using Stochastic Context Free Grammars (SCFGs) have been developed to compare two RNA structures. Original SCFG models [56, 57] require a prior multiple sequence alignment (with structure annotation) for the training purpose, thus their applicability is limited to RNA types for which structures of a large number of sequences are available, such as snoRNA and tRNA [56,58]. However, Rsearch [59] and stemloc [60], both based on SCFG, are capable of conducting pair-wise structure comparisons with no requirement for pre-alignment. Rsearch uses RIBOSUM substitution matrices derived from ribosomal RNAs to score the matches in single-stranded (ss) and double-stranded (ds) regions. stemloc uses "fold envelope" to improve efficiency by confining the search space involved in calculations. The time and space complexities of these two tools are also listed in Table 3.1. Furthermore, pattern-based techniques such as RNAmotif, RNAmot and PatSearch [22, 37, 61] have been used in database searches to detect similar RNA substructures. These tools represent RNA structures by a consensus pattern containing both sequence and structure information. One important advantage of these pattern-based tools is the ability of dealing with pseudoknots.

This thesis presents a computationally efficient tool, called RSmatch, capable of both globally and locally aligning two RNA secondary structures. RSmatch does not require any prior knowledge of structures of interest. It can uncover structural similarities by means of direct aligning at the structure level. Its application to database search and multiple alignment are also demonstrated. Furthermore, RSmatch was compared with three widely used tools, PatSearch [25], stemloc [60] and Rsearch [59]. It showed that RSmatch was faster or achieved comparable or higher accuracy than the existing tools when applied to a number of known RNA structures, including simple stem loops and complex structures containing junctions.

CHAPTER 4

ALGORITHM TO ALIGNING RNA SECONDARY STRUCTURES

4.1 Secondary Structure Decomposition

RSmatch models RNAs by a structure decomposition scheme similar to the loop model commonly used in the algorithms for RNA structure prediction [27, 28]. With this model, pseudoknots are not allowed. This method differs from the loop decomposition methods in that it completely decomposes an RNA secondary structure into units called circles (Figure 4.1A). When the secondary structure is depicted on a plane, a circle is defined as a set of nucleotides that are reachable from one another without crossing any base pair. As shown in Figure 4.1A, all circles are closed or ended by a base pair except the first circle (circle one in the Figure 4.1A), which always contains the $5'$-most and the $3'$-most bases. Various types of RNA structures, such as bulge, loop, and junction can be represented by circles, as shown in Figure 4.1A.

Circles of an RNA structure can be organized as a hierarchical tree according to their relative positions in the secondary structure, where each tree node corresponds to a circle (Figure 4.1B). This tree organization is informative to deduce the structural relationship among circles and reflects the structure particularities of the given RNA secondary structure. If two circles reside on the same lineage (path) in the tree, the circle appearing higher in the tree is called an ancestor of the other, and the latter is a descendent of the former. As a result, in the context of the hierarchical tree, two distinct circles fall into one of the following two categories, in the order of decreasing closeness: (i) the two circles maintain an ancestor/descendent relationship, or (ii) they share a common ancestor in the tree. For example, in Figure 4.1B, circle 2 is an ancestor of circle 5, whereas circle 6 does not have ancestor/descendent relationship with circle 5 since they are not on the same lineage. The double-stranded region or stem of a structure is decomposed into a set of "degenerated" circles, each containing

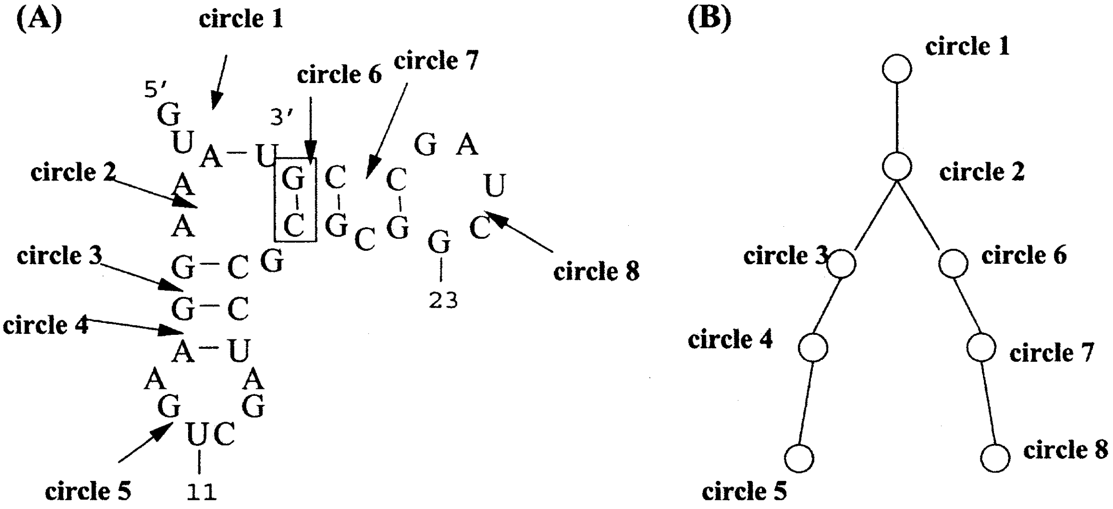


Figure 4.1 RNA structure decomposition into circles. (A) A hypothetical secondary structure is decomposed into a set of circles by circle decomposition procedure; (B) Decomposed circles are organized into a hierarchy.

only two base pairs. As such, a stem of n bases in length will result in $n-1$ consecutive degenerated circles. Since a base pair may have two associated circles; one circle is named as "the parent circle" and the other "the child circle" according to their positions in the hierarchical tree. For example, for the boxed C-G base pair in Figure 4.1A, circle 2 is its parent circle and circle 6 is its child circle.

4.2 Structure Alignment Formalization

Given an RNA secondary structure, consider two types of structure components, single bases and base pairs, in the secondary structure. To integrate both sequence and structure information, two constraints are introduced among the structure components: precedence constraint and hierarchy constraint. The precedence constraint is defined as the precedence order among structure components and the hierarchy constraint specifies the inter-component relationship in the context of the hierarchical tree described above. The precedence order is determined by the $3'$ bases of individual structure components: the one with its $3'$ base closer to the RNA sequence's $5'$-end precedes the other. For example, in Figure 4.1A, the single base component

U (marked as the 11th nucleotide) in circle 5 precedes the base pair component **C-G** (boxed) in circle 6.

To capture the inter-component relationship within the hierarchical tree context, it is needed to map each structure component to a circle in the tree. It is obvious that each single base can be mapped to a unique circle. However, a base pair could be mapped to two alternate circles: one parent circle and one child circle. To resolve this ambiguity, it is always required a component be mapped to its parent circle. The inter-component relationship is then reduced to the inter-circle relationship of three types: (i) ancestor/descendent, (ii) common ancestor, and (iii) identical circle.

Given two RNA secondary structures A and B, where A, referred to as the query structure, has m structure components A^1, A^2, ..., A^m and B, referred to as the subject structure, has n structure components B^1, B^2, ..., B^n, the structure alignment between A and B is formalized as a conditioned optimization problem based on the above two constraints: given a scoring scheme consisting of two matrices, one for matching two single bases and the other for matching two base pairs, find an optimal alignment between the two sets of structure components such that the aforementioned precedence and hierarchy constraints are preserved for any two matched component pairs (A^i, B^i) and (A^j, B^j). In other words, the two structure constraints between A^i and A^j must be respectively equivalent to that between B^i and B^j. This formalization has an implicit biological significance in that a single stranded region in one structure, if not aligned to a gap as a whole, will always align with a single stranded region in the other structure. This alignment requirement is important because single stranded regions are usually treated as functional units in binding to specific proteins.

4.3 Algorithmic Framework

A dynamic programming algorithm is employed in RSmatch. As with sequence alignment, the structure alignment could be either global or local. The difference lies only in the setup of initialization conditions; the algorithmic framework is the same since both global and local alignment must preserve the two constraints described above.

A scoring table is established with its rows/columns corresponding to structure components of the two given RNA secondary structures. The rows/columns are organized in such a way that the precedence and hierarchy constraints are combined and easy to follow in the course of alignment computation. Specifically, the structure components of each structure are sorted according to the precedence order defined above. It is easy to see that this arrangement of rows/columns makes the precedence constraint automatically preserved. However, preservation of the hierarchy constraint is much more complicated and can only be accomplished in the derivative analysis for each cell (entry) in the scoring table. The detailed derivation will be be discussed in the course of filling in the scoring table.

Each cell of the scoring table represents an intermediate comparison between two partial structures corresponding to the cell's row and column components (either single base or base pair) respectively. The partial structure with respect to a structure component c (single base or base pair) is a set of structure components S_c such that for any component $a \in S_c$, the following three structure constraints between c and a must be satisfied: (i) a precedes c; (ii) by the hierarchy constraint, a is not an ancestor of c; and (iii) c itself is included in S_c.

Furthermore, since a base pair could appear in two circles, its corresponding partial structure could be divided into two smaller substructures: parent structure and child structure. Formally, given a base pair component c, the parent structure of c is the set of structure components $P_c \subseteq S_c$ (excluding c itself) such that for

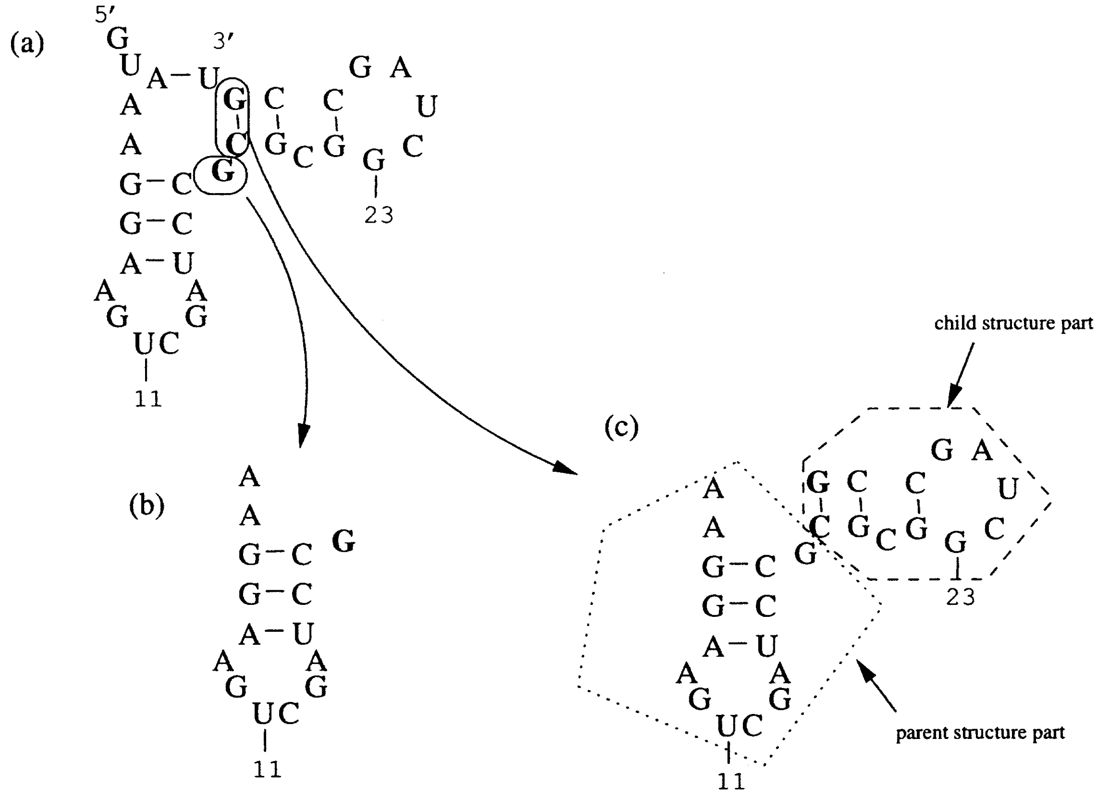


Figure 4.2 Partial structure determination. (a) A hypothetical RNA structure is used to illustrate how partial structures are determined. (b) The partial structure for a single base G in boldface is shown. (c) The partial structure for the base pair C-G in boldface consists of two parts, a parent structure part and a child structure part. The base pair itself is included in the child structure.

any component $a \in P_c$, a's $3'$-base is always $5'$ upstream of c's $5'$-base; the child structure of c is the set of structure components $L_c \subseteq S_c$ (including c) such that for any component $a \in L_c$, a's $5'$-base is always $3'$ downstream of c's $5'$-base. It can be shown that $P_c \cup L_c = S_c$ and $P_c \cap L_c = \phi$. Examples of partial structures are given in Figure 4.2. As shown, for a base pair, its child and parent structures together constitute the whole partial structure for the base pair.

In the following discussions, the concept of a partial structure and its byproducts (parent structure and child structure) form the kernel of the proposed algorithmic framework. As such, the RNA structure alignment problem can be progressively solved by aligning small structures and expanding each of them one component at a time until all structure components are covered.

4.4 Preliminaries

Cells in the scoring table are processed row by row from top to bottom and from left to right within each row. By considering the row/column components, three types of cells are considered: (i) a cell corresponding to two single bases; (ii) a cell corresponding to one single base and one base pair; and (iii) a cell corresponding to two base pairs. For (i), each cell stores the score of aligning the partial structures corresponding to the cell's row and column components respectively. For (ii) and (iii), it is needed to consider alignments involving the partial and child structures induced by the base pair components. Notice that the parent structures of the base pair components are excluded. It can be shown that each parent structure P_c of component c can always be considered as the partial structure S_x of some other component x, which means it is only needed to consider child and partial structures in the alignment computation, excluding the parent structure. Consequently, the above three types of cells have one, two and four alignment scores respectively.

A scoring scheme is required to score the match of two structure components. Here the scoring scheme is defined as a function $g(a, b)$ where a and b represent two structure components that are matched with each other. Another important aspect of the alignment algorithm is to penalize the match involving gap(s). In the course of computation, one structure component (single base or base pair) could match with a gap or a whole small structure (parent or child structure) could match with a large gap. Intuitively, the larger the gap is, the heavier the penalty will be. In the implementation, an atomic penalty value was adopted, denoted as u, for the smallest gap equivalent to a single base. The penalty value for a large gap is proportional to its size in terms of the number of bases matched with the gap.

Let A^* be a small structure in the query RNA structure A and B^* a small structure in the subject RNA structure B. The score obtained by aligning the two structures A^* and B^*, denoted as $f(A^*, B^*)$, is:

$$f(A^*, B^*) = \sum_{\substack{a \in A^* \\ b \in B^*}} g(a, b) + u \cdot G$$

where G represents the total number of gaps in aligning A^* and B^*.

4.5 Initialization

Assume that the row components ($a^{'}$s) are from the query RNA structure A and the column components ($b^{'}$s) from the subject RNA structure B. Here the focus is on global alignment while the initializations for local alignment can be derived similarly. The initialization conditions deal with the cases where at least one of the structures under alignment is an empty structure ϕ. This is equivalent to setting up the 0th row/column in the scoring table. As discussed above, each base pair component has two small structures to be considered: a child structure and a partial structure. Thus, the aforementioned three types of cells have one, two and four initialization scores respectively.

For a given structure component x (single base or base pair), let S_x represent its partial structure. If x is a base pair, L_x denotes its child structure. It is obvious that $f(\phi, \phi) = 0$. Furthermore, for any structure components a and b, $f(S_a, \phi) = |S_a| \cdot u$, $f(\phi, S_b) = |S_b| \cdot u$; if a and b are base pairs, $f(L_a, \phi) = |L_a| \cdot u$ and $f(\phi, L_b) = |L_b| \cdot u$ where $| \cdot |$ represents the cardinality of the respective set.

4.5.1 Filling in the Scoring Table

The simplest cell type is the one with both row and column components are a single bases a, b. Let a^p denote the structure component that precedes a by precedence order established before. Formally, in matching the partial structure S_a with the partial structure S_b there are only three possibilities: (i) a is aligned with b; (ii) a is aligned with a gap; and (iii) b is aligned with a gap. Thus the score of matching S_a with S_b can be calculated by Eq.4.1.

$$f(S_a, S_b) = \max \begin{cases} f(S_{a^p}, S_{b^p}) + g(a, b) \\ f(S_{a^p}, S_b) + \mu \\ f(S_a, S_{b^p}) + \mu \end{cases} \tag{4.1}$$

The second cell type is the one formed by one single base and one base pair. There are actually two symmetric subtypes where either a or b is a base pair. Since the analysis is identical, the former case where a is a base pair is discussed. As discussed before, besides the partial structure S_a it is needed to consider the child structure L_a derived from the base pair a. Thus, for this type of cells, two alignment scores need to be computed.

By the principle of dynamic programming, the smaller size problem needs to be solved before the larger size problem. Thus the structure alignment between the child structure L_a and the partial structure S_b is calculated first. There are only two possibilities: (i) the single base component b is aligned with a gap; and (ii) the base pair a is aligned with a gap (see Figure 4.3a), which is summarized by Eq.4.2:

$$f(L_a, S_b) = \max \begin{cases} f(L_a, S_{b^p}) + \mu \\ f(S_{a^p}, S_b) + 2\mu \end{cases} \tag{4.2}$$

In aligning the partial structure S_a with the partial structure S_b, to preserve precedence and hierarchy constraints simultaneously, there are only three possibilities: (i) the single base b matches with a gap; (ii) the partial structure S_b matches with the

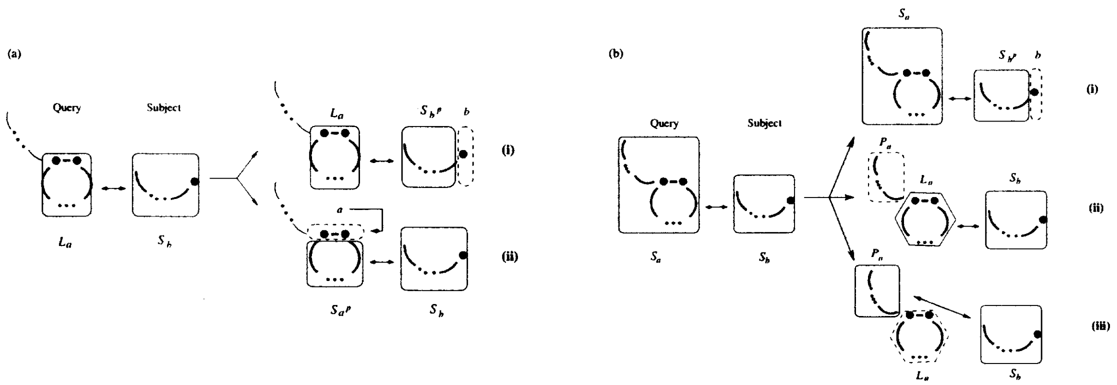


Figure 4.3 Alignment derivation involving two partial structures where one is derived from a base pair (query) while the other from a single base component (subject). The nucleotide bases are depicted as big dots and short dash connecting two dots represents bond. The corresponding cell contains two values: one is the alignment score concerning the base pair's *child structure* (a); the other concerning with the whole partial structure (b). For (a), the optimal alignment is obtained from either matching the subject structure's single base to a gap (a.i) or matching the query structure's base pair to a gap (a.ii). For (b), the optimal alignment is derived from one of the three possibilities: matching the subject structure's single base to a gap (b.i), matching query structure's *child structure* to the whole partial structure at subject side (b.ii), or matching query structure's *parent structure* to the whole partial structure at subject side (b.iii).

child structure L_a ; (iii) the partial structure S_b matches with the parent structure P_a (see Figure 4.3b). Thus, the Eq.4.3 is as:

$$f(S_a, S_b) = \max \begin{cases} f(S_a, S_{b^P}) + \mu \\ f(L_a, S_b) + |P_a| \cdot \mu \\ f(P_a, S_b) + |L_a| \cdot \mu \end{cases} \tag{4.3}$$

For the third cell type, a is a base pair and b is also a base pair. It is needed to compute four alignment scores because each base pair component contributes two structures: one child structure and one partial structure. While aligning the child structure L_a with the child structure L_b, it is clear that:

$$f(L_a, L_b) = \max \begin{cases} f(S_{a^p}, S_{b^p} + g(a, b) \\ f(S_{a^p}, L_b) + 2\mu \\ f(L_a, S_{b^p}) + 2\mu \end{cases} \tag{4.4}$$

since both a and b are the last components in the respective child structures by precedence order.

The following equation (Eq.4.5 gives the alignment score between the partial structure S_a and the child structure L_b:

$$f(S_a, L_b) = \max \begin{cases} f(S_a, S_{b^p}) + 2\mu \\ f(P_a, L_b) + |L_a| \cdot \mu \\ f(L_a, L_b) + |P_a| \cdot \mu \end{cases} \tag{4.5}$$

The first case corresponds to that b is aligned with a gap. If b does not match with a gap, it can be shown that, to preserve both precedence and hierarchy constraints, the second and third cases in the above equation cover all possible situations. Similarly, the score of aligning the child structure L_a and the partial structure S_b can be calculated as shown in Eq.4.6:

$$f(L_a, S_b) = \max \begin{cases} f(S_{a^p}, S_b) + 2\mu \\ f(L_a, P_b) + |L_b| \cdot \mu \\ f(L_a, L_b) + |P_b| \cdot \mu \end{cases} \tag{4.6}$$

In aligning the partial structure S_a with the partial structure S_b, there are five possibilities: (i) the parent structure P_a is matched with the parent structure P_b and the child structure L_a is matched with the child structure L_b; (ii) the child structure L_a is matched with gaps; (iii) the child structure L_b is matched with gaps; (iv) the

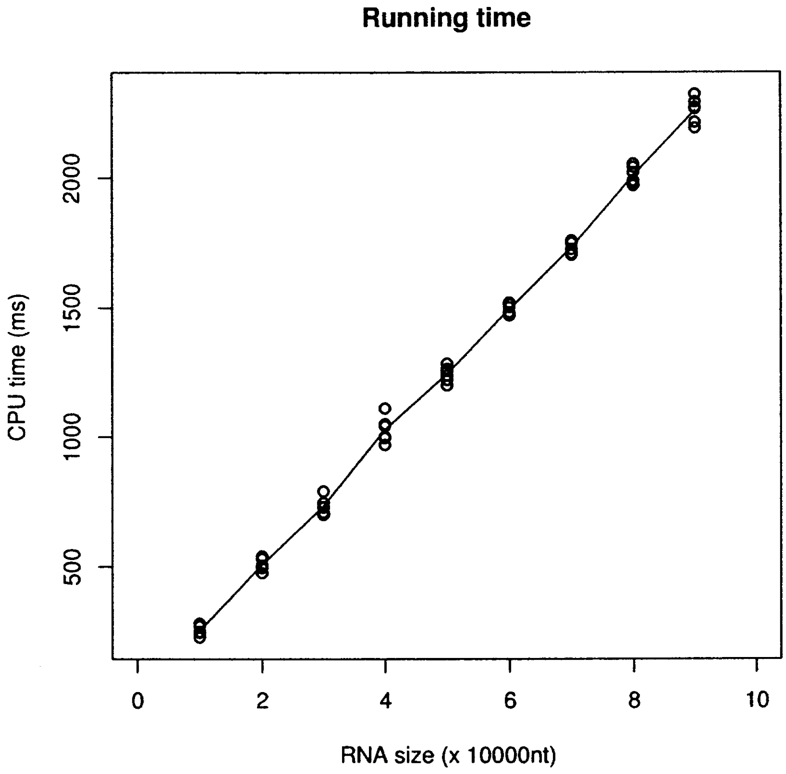


Figure 4.4 CPU time versus RNA size. The averaged time points are connected. The nearly perfect linear growth of the running time with respect to the RNA size gives obvious experimental proof that the algorithm's time complexity is bounded by $\mathcal{O}(n^2)$ where n is the number of structure components.

parent structure P_a is matched with gaps; and (v) the parent structure P_b is matched with gaps. Therefore:

$$f(S_a, S_b) = \max \begin{cases} f(P_a, P_b) + f(L_a, L_b) \\ f(P_a, S_b) + |L_a| \cdot \mu \\ f(S_a, P_b) + |L_b| \cdot \mu \\ f(L_a, S_b) + |P_a| \cdot \mu \\ f(S_a, L_b) + |P_b| \cdot \mu \end{cases} \tag{4.7}$$

4.6 Running Efficiency

By dynamic programming, the running time of computing an alignment equals the number of writing operations needed to fill the scoring table. Thus the time complexity of RSmatch is $\mathcal{O}(mn)$, where m (n, respectively) is the number of structure components in the query (subject, respectively) RNA structure. To test the scalability, the seed sequences for 5S rRNA family was downloaded from Rfam and one annotated structure was randomly selected as the query while folding the rest sequences were folded to prepare the structure database as discussed in the next chapter. The RSmatch running time versus the database size was plotted. The program was run 10 times and the result was shown in Figure 4.4. The nearly perfect linear growth of the running time gives an empirical proof that the algorithm's time complexity is bounded by $\mathcal{O}(nm)$.

CHAPTER 5

APPLICATION TO RNA MOTIF DETECTION

5.1 Data Set

All experiments (unless otherwise specified) were carried out on a Linux system with two 2.4 GHz Intel processors and 3 GB memory. A human UTR structure database was constructed as follows. 19,986 human RefSeq mRNA sequences (January 2004 version) were downloaded from National Center for Biotechnology Information (NCBI). Each RefSeq sequence containing UTR regions as indicated by RefSeq's GenBank annotation, was processed to extract its $5'$UTR and $3'$UTR sequences. For each UTR sequence, $100nt$ subsequence was taken at every $50th$ nucleotide position from $5'$ to $3'$, making consecutive subsequences overlap with one another on a $50nt$ segment. Subsequences shorter than $100nt$, e.g. at the $3'$ end, were also kept. Using the Vienna RNA package's RNAsubopt function with setting "-e 0", all sequences were folded to form the structure database. For any given RNA sequence, the setting "-e 0" will generate multiple RNA structures all having the minimum free energy. The final database consisted of 575,000 RNA secondary structures.

The structural patterns of a histone $3'$UTR stem-loop structure (HSL3) and an iron responsive element (IRE) were used in this study, based on their specifications in the UTRdb database [37]. Three tools, PatSearch [25], stemloc [60] and Rsearch [59], were employed for comparison purposes. The efficiency of these tools was measured by CPU running time. The performance of each program was assessed by specificity and sensitivity. Specificity was calculated as TP/(TP + FP) and sensitivity as TP/(TP + FN), where TP was the number of true positives, FP the number of false positives, and FN the number of false negatives.

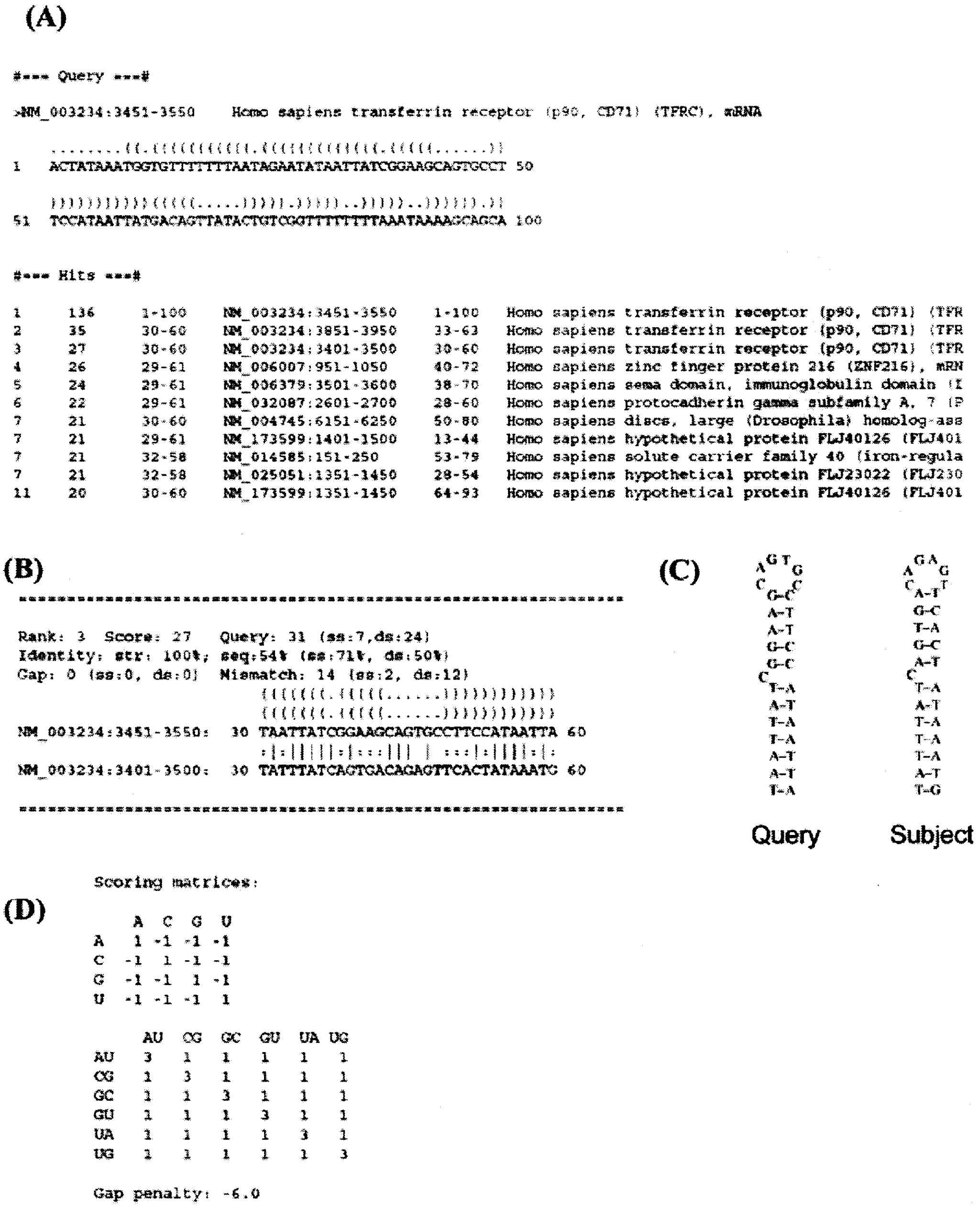
To test the applicability of RSmatch to complex structures, RNA families were downloaded from Rfam [35]. Only chose families that have more than 10 seed RNAs and the consensus sequence length longer than 250 nucleotides are chosen in this study. The final data set has 64 families. For each family, one member RNA was randomly selected as the query RNA and its structure was obtained from Rfam. Another set of 10 subject RNAs in the same family was then randomly chosen. Here noise was intentionally introduced by extending each subject RNA sequence with its adjacent sequences at both $3'$ and $5'$ ends to make the total length three times of its original one.

5.1.1 Performance on Stem-loop Structures

Using RSmatch, RNA motifs in UTR regions of human mRNA sequences were first studied. A well-known fact is that the accuracy and efficiency of RNA folding programs will decrease significantly when the sequences to be folded become very long. Satisfactory performance is usually obtained when the sequences have moderate lengths, *i.e.* one hundred nucleotides. Thus, a moving window scheme was used to get subsequences of $100nt$ and fold them using the Vienna RNA package [39]. In the RSmatch package, this subsequence length is a user-defined parameter.

Since the nucleotide conservation in the single-stranded region of an RNA sequence may differ from that in the double-stranded region, two scoring matrices were used, one for substitutions among single bases and the other among base pairs. This type of scoring scheme was also used in other studies [10, 59]. Theoretically, the scoring matrix for single bases is a 4×4 table for all types of substitutions of single nucleotides, and the one for base pairs is a 16×16 table for all types of substitutions of base pairs. However, using Vienna RNA package, only six types of base pairs were observed, *i.e.* Watson-Crick base pairs A-U, U-A, G-C, C-G, and wobble base pairs G-U and U-G. Values used in the two matrices were empirically chosen so as to conform to the general understanding of the sequence and structure conservation of RNA motifs, as follows. (1) Mutations in the double-stranded region may not be detrimental to RNA's function if the mutated sequence still preserves the same secondary structure. Therefore base pair substitutions were rewarded with a positive score, instead of a penalty. (2) A sequence in the single-stranded region may be important for RNA's function, such as binding to proteins, and thus mismatches were penalized. An simplified function $u \times l$ was utilized to process gaps, where u was the atomic penalty value for a gap which is one single base long and l is the length of the gap in terms of the number of bases matched with the gap. In the experiments otherwise stated explicitly, the u was empirically set to -6 and changing the u value did not change the qualitative conclusion made provided that the absolute value of u was greater than any positive score in the scoring matrices. Users can freely change the u value when applying RSmatch to their own data set.

RSmatch was first tested using a query sequence containing an iron response element (IRE). The IRE motif is a bipartite stem-loop structure containing 30 nucleotides. Two alternative types of IREs have been found, which differ in the middle region [37]. Type I has a bulge, whereas type II has a small internal loop. IREs have been found in both $5'$ and $3'$ UTRs of genes that are involved in iron homeostasis in higher eukaryotic species. They interact with iron regulatory proteins (IRPs) and play key roles in RNA stability and translation. Using a subsequence in the $3'$UTR of transferrin receptor (NM_003234) that contains an IRE motif, database search was conducted against the UTR structure database described above. A list of top hits is shown in Fig.5.1. The best hit of the search is the query structure itself, as expected. Other regions of the same mRNA and regions of other RNAs are also found to have homologous structures with the query. As clearly shown in the result, the region containing the IRE motif, which is from about the $30th$ nucleotide to about the $60th$ nucleotide of the query structure, has been located by the RSmatch program, indicating that a local optimal alignment has been achieved. Among the top 10 hits, several sequences are known to have IREs, such as several regions in the $3'$UTR of



(A)

```
#=== Query ===#

>NM_003234:3451-3550    Homo sapiens transferrin receptor (p90, CD71) (TFRC), mRNA

    ........((.(((((((((((.(((((((((((((.(((((......))
1   ACTATAAATGGTGTTTTTTTAATAGAATATAATTATCGGAAGCAGTGCCT 50

    ))))))))))))(((((.....))))).)))))..)))))..))))).))
51  TCCATAATTATGACAGTTATACTGTCGGTTTTTTTTAAATAAAAGCAGCA 100


#=== Hits ===#

1    136    1-100    NM_003234:3451-3550    1-100    Homo sapiens transferrin receptor (p90, CD71) (TFR
2    35     30-60    NM_003234:3851-3950    33-63    Homo sapiens transferrin receptor (p90, CD71) (TFR
3    27     30-60    NM_003234:3401-3500    30-60    Homo sapiens transferrin receptor (p90, CD71) (TFR
4    26     29-61    NM_006007:951-1050     40-72    Homo sapiens zinc finger protein 216 (ZNF216), mRN
5    24     29-61    NM_006379:3501-3600    38-70    Homo sapiens sema domain, immunoglobulin domain (I
6    22     29-61    NM_032087:2601-2700    28-60    Homo sapiens protocadherin gamma subfamily A, 7 (P
7    21     30-60    NM_004745:6151-6250    50-80    Homo sapiens discs, large (Drosophila) homolog-ass
7    21     29-61    NM_173599:1401-1500    13-44    Homo sapiens hypothetical protein FLJ40126 (FLJ401
7    21     32-58    NM_014585:151-250      53-79    Homo sapiens solute carrier family 40 (iron-regula
7    21     32-58    NM_025051:1351-1450    28-54    Homo sapiens hypothetical protein FLJ23022 (FLJ230
11   20     30-60    NM_173599:1351-1450    64-93    Homo sapiens hypothetical protein FLJ40126 (FLJ401
```

(B)

```
==================================================================

Rank: 3  Score: 27   Query: 31 (ss:7,ds:24)
Identity: str: 100%; seq:54% (ss:71%, ds:50%)
Gap: 0 (ss:0, ds:0)  Mismatch: 14 (ss:2, ds:12)
                            (((((((.(((((......))))))))))))
                            (((((((.(((((......))))))))))))
NM_003234:3451-3550:  30 TAATTATCGGAAGCAGTGCCTTCCATAATTA 60
                         :|:|||||:|:::|||  |  :::|:||||:|:
NM_003234:3401-3500:  30 TATTTATCAGTGACAGAGTTCACTATAAATG 60

==================================================================
```

(C)

(D)

```
Scoring matrices:

     A  C  G  U
A    1 -1 -1 -1
C   -1  1 -1 -1
G   -1 -1  1 -1
U   -1 -1 -1  1

     AU  CG  GC  GU  UA  UG
AU   3   1   1   1   1   1
CG   1   3   1   1   1   1
GC   1   1   3   1   1   1
GU   1   1   1   3   1   1
UA   1   1   1   1   3   1
UG   1   1   1   1   1   3

Gap penalty: -6.0
```

Figure 5.1 Database search with an RNA structure containing an IRE motif. A structure element in the 3′UTR of human transferrin receptor (NM_003234) was used as a query to search the UTR structure database. (A) The output from RSmatch showing the top 11 hits. The six columns in the "Hits" section are, from left to right, rank, alignment score, region in the query, name of the hit, region in the hit, and annotation of the hit respectively. (B) A pairwise alignment of the query structure and a hit structure (NM_003234:3401-3500). (C) The RNA structures corresponding to the query and the subject (hit) structure in (B). (D) Scoring matrices and the gap penalty used in the search. T and U are used interchangeably in this study.

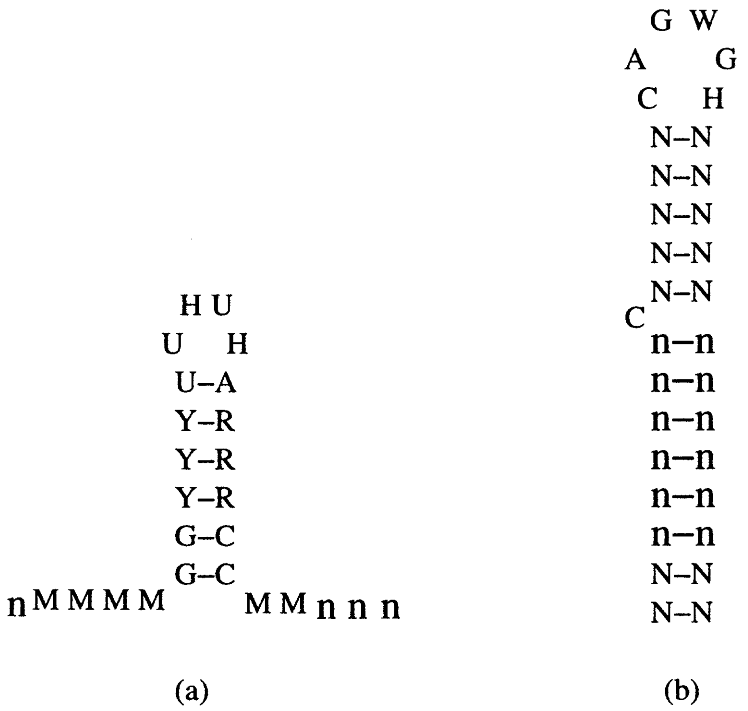


Figure 5.2 The two pattern-based RNA structures used in this study. (a) Histone 3′-UTR (HSL3) motif. (b) Iron Response Element (IRE) motif. A wildcard, represented by a lowercase letter *n*, is allowed to appear in a motif where the length of a region, either single-stranded or double-stranded is a variable. For example, the 5′ flanking tail of HSL3 can be 4 or 5 nt long, and the lower part of the stem region of IRE can be 2 to 8 nt long.

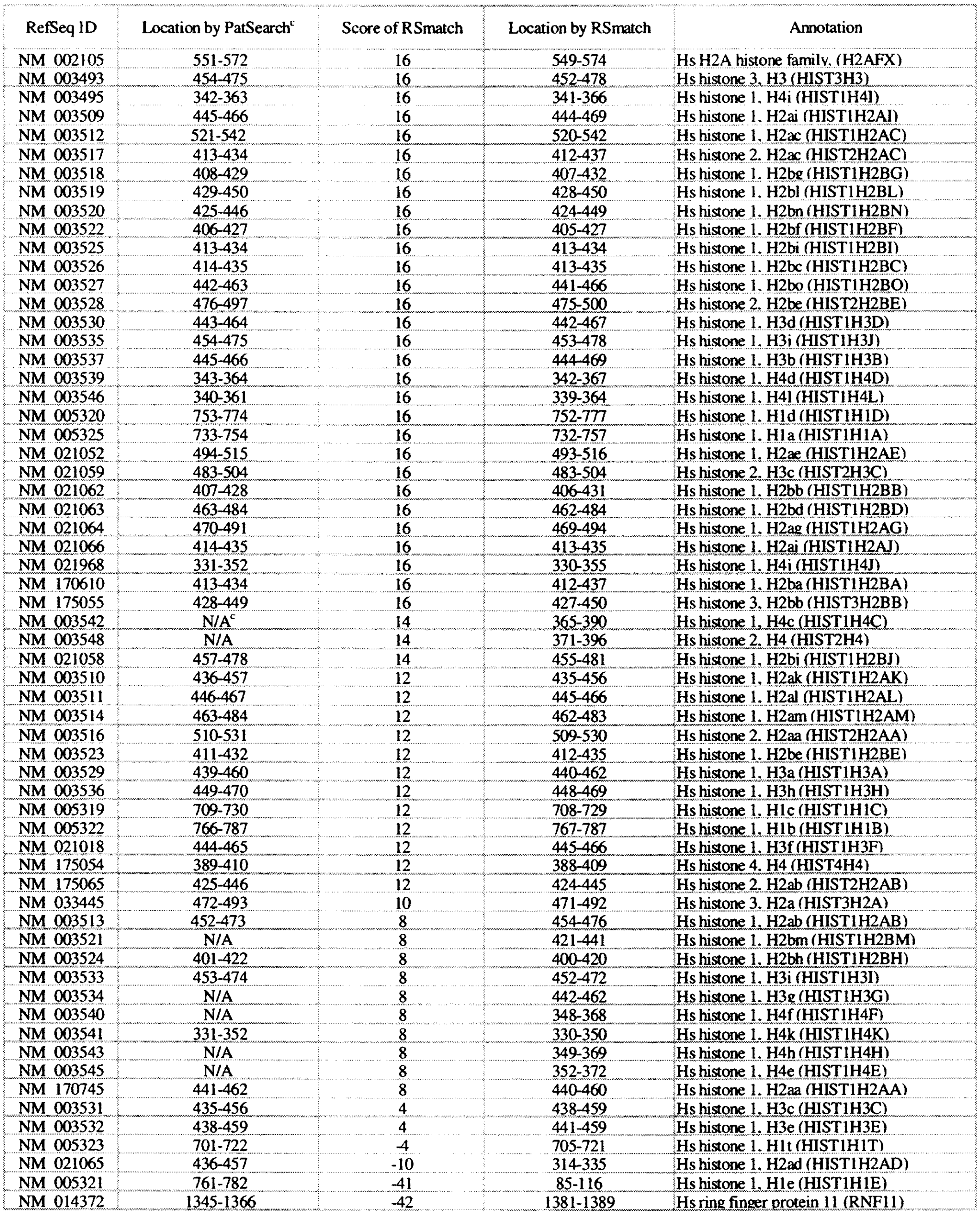
transferrin receptor (NM_003234) and the 5′UTR of solute carrier family 40 protein (NM_014585). Other top hits have not been shown so far to have IREs. It is not known if some of them are novel IRE-containing RNAs and the definitive answer will await wet lab validation. The output shows detailed alignment and related information, including the numbers of bases in the single-stranded and double-stranded regions, and the percentages of identity in single-stranded and double-stranded regions.

RSmatch can also accept pattern-based RNA structures (also called pattern descriptors) to search a structure database. Since a pattern-based search method has an intrinsic primitive scoring scheme by using degenerate bases, simplified binary matrices were used as the equivalent to score an alignment. In the matrices, the match of a pair of structure components (single bases or base pairs), including those containing degenerated bases, was given a score of 1, a mismatch was penalized by a score of -1, and the atomic gap penalty u was set to -3.

To allow variability in single-stranded and/or double-stranded regions for a structure pattern, a wildcard "n"(lower case) was introduced to represent optional single base component ("n") and base pair component ("n-n"). The meaning of "n" is identical to the IUB code "N" except that the matching score for both structure components "n" and "n-n" is always zero regardless of whether they are aligned with a structure component or a gap. Two RNA motifs, namely a histone $3'$UTR stem-loop structure (HSL3) and IRE, were used to test the proposed method. HSL3, which resides in the $3'$UTR region of histone mRNAs, has a typical stem loop structure with two flanking tails (Fig.5.2(a)). Both the stem and the flanking sequences are important to bind with a stem-loop binding protein (SLBP), which controls the pre-mRNA processing and stability of histone mRNAs [38]. In contrast to the HSL3 motif, IRE is relatively flexible in length and in nucleotide composition in its stem region (Fig.5.2(b)). RSmatch was compared with PatSearch [25], a widely used tool that searches a sequence database for sequence and structure patterns.

Using the HSL3 motif and UTR sequence database, PatSearch found 55 hits whose locations were presented in Table 5.1. Among them, one is a false positive (NM_014372, ring finger protein 11, shown at the bottom of Table 5.1). Therefore the specificity (98.2%) of PatSearch is very high. This is attributable to the precise specification of the HSL3 pattern. However, if a pattern description is too precise, it may lead to the "overfitting" problem. This problem prevents the tool from finding slightly divergent structures, thus lowering the tool's sensitivity. Indeed, several histone genes were not detected by PatSearch, including two histone genes (histone H4c NM_003542 and histone H4 NM_003548) which were found by RSmatch among its top 33 hits.

Table 5.1 HSL3 motifs found by RSmatch and PatSearch[a,b]



RefSeq ID	Location by PatSearch[c]	Score of RSmatch	Location by RSmatch	Annotation
NM 002105	551-572	16	549-574	Hs H2A histone family, (H2AFX)
NM 003493	454-475	16	452-478	Hs histone 3, H3 (HIST3H3)
NM 003495	342-363	16	341-366	Hs histone 1, H4i (HIST1H4I)
NM 003509	445-466	16	444-469	Hs histone 1, H2ai (HIST1H2AI)
NM 003512	521-542	16	520-542	Hs histone 1, H2ac (HIST1H2AC)
NM 003517	413-434	16	412-437	Hs histone 2, H2ac (HIST2H2AC)
NM 003518	408-429	16	407-432	Hs histone 1, H2bg (HIST1H2BG)
NM 003519	429-450	16	428-450	Hs histone 1, H2bl (HIST1H2BL)
NM 003520	425-446	16	424-449	Hs histone 1, H2bn (HIST1H2BN)
NM 003522	406-427	16	405-427	Hs histone 1, H2bf (HIST1H2BF)
NM 003525	413-434	16	413-434	Hs histone 1, H2bi (HIST1H2BI)
NM 003526	414-435	16	413-435	Hs histone 1, H2bc (HIST1H2BC)
NM 003527	442-463	16	441-466	Hs histone 1, H2bo (HIST1H2BO)
NM 003528	476-497	16	475-500	Hs histone 2, H2be (HIST2H2BE)
NM 003530	443-464	16	442-467	Hs histone 1, H3d (HIST1H3D)
NM 003535	454-475	16	453-478	Hs histone 1, H3i (HIST1H3J)
NM 003537	445-466	16	444-469	Hs histone 1, H3b (HIST1H3B)
NM 003539	343-364	16	342-367	Hs histone 1, H4d (HIST1H4D)
NM 003546	340-361	16	339-364	Hs histone 1, H4l (HIST1H4L)
NM 005320	753-774	16	752-777	Hs histone 1, H1d (HIST1H1D)
NM 005325	733-754	16	732-757	Hs histone 1, H1a (HIST1H1A)
NM 021052	494-515	16	493-516	Hs histone 1, H2ae (HIST1H2AE)
NM 021059	483-504	16	483-504	Hs histone 2, H3c (HIST2H3C)
NM 021062	407-428	16	406-431	Hs histone 1, H2bb (HIST1H2BB)
NM 021063	463-484	16	462-484	Hs histone 1, H2bd (HIST1H2BD)
NM 021064	470-491	16	469-494	Hs histone 1, H2ag (HIST1H2AG)
NM 021066	414-435	16	413-435	Hs histone 1, H2aj (HIST1H2AJ)
NM 021968	331-352	16	330-355	Hs histone 1, H4i (HIST1H4J)
NM 170610	413-434	16	412-437	Hs histone 1, H2ba (HIST1H2BA)
NM 175055	428-449	16	427-450	Hs histone 3, H2bb (HIST3H2BB)
NM 003542	N/A[c]	14	365-390	Hs histone 1, H4c (HIST1H4C)
NM 003548	N/A	14	371-396	Hs histone 2, H4 (HIST2H4)
NM 021058	457-478	14	455-481	Hs histone 1, H2bj (HIST1H2BJ)
NM 003510	436-457	12	435-456	Hs histone 1, H2ak (HIST1H2AK)
NM 003511	446-467	12	445-466	Hs histone 1, H2al (HIST1H2AL)
NM 003514	463-484	12	462-483	Hs histone 1, H2am (HIST1H2AM)
NM 003516	510-531	12	509-530	Hs histone 2, H2aa (HIST2H2AA)
NM 003523	411-432	12	412-435	Hs histone 1, H2be (HIST1H2BE)
NM 003529	439-460	12	440-462	Hs histone 1, H3a (HIST1H3A)
NM 003536	449-470	12	448-469	Hs histone 1, H3h (HIST1H3H)
NM 005319	709-730	12	708-729	Hs histone 1, H1c (HIST1H1C)
NM 005322	766-787	12	767-787	Hs histone 1, H1b (HIST1H1B)
NM 021018	444-465	12	445-466	Hs histone 1, H3f (HIST1H3F)
NM 175054	389-410	12	388-409	Hs histone 4, H4 (HIST4H4)
NM 175065	425-446	12	424-445	Hs histone 2, H2ab (HIST2H2AB)
NM 033445	472-493	10	471-492	Hs histone 3, H2a (HIST3H2A)
NM 003513	452-473	8	454-476	Hs histone 1, H2ab (HIST1H2AB)
NM 003521	N/A	8	421-441	Hs histone 1, H2bm (HIST1H2BM)
NM 003524	401-422	8	400-420	Hs histone 1, H2bh (HIST1H2BH)
NM 003533	453-474	8	452-472	Hs histone 1, H3i (HIST1H3I)
NM 003534	N/A	8	442-462	Hs histone 1, H3g (HIST1H3G)
NM 003540	N/A	8	348-368	Hs histone 1, H4f (HIST1H4F)
NM 003541	331-352	8	330-350	Hs histone 1, H4k (HIST1H4K)
NM 003543	N/A	8	349-369	Hs histone 1, H4h (HIST1H4H)
NM 003545	N/A	8	352-372	Hs histone 1, H4e (HIST1H4E)
NM 170745	441-462	8	440-460	Hs histone 1, H2aa (HIST1H2AA)
NM 003531	435-456	4	438-459	Hs histone 1, H3c (HIST1H3C)
NM 003532	438-459	4	441-459	Hs histone 1, H3e (HIST1H3E)
NM 005323	701-722	-4	705-721	Hs histone 1, H1t (HIST1H1T)
NM 021065	436-457	-10	314-335	Hs histone 1, H2ad (HIST1H2AD)
NM 005321	761-782	-41	85-116	Hs histone 1, H1e (HIST1H1E)
NM 014372	1345-1366	-42	1381-1389	Hs ring finger protein 11 (RNF11)

[a] Items listed here include those found by PatSearch and those found by RSmatch using cutoff value of 8 that are related to histone genes;

[b] RSmatch gets 33 hits at cutoff value of 14 and gets 184 hits at cutoff value of 8;

[c] mRNAs that are not detected to have the HSL3 motif by PatSearch are marked with "N/A".

Several other histone genes appeared among the top 184 hits of RSmatch (Table 5.1). This indicates that by gaining specificity, PatSearch loses sensitivity for HSL3. Since RSmatch gives a score to each alignment, different cutoffs can be used for selecting top hits (Table 5.2). It seems that newly detected true positives are heavily outnumbered by false positives as RSmatch relaxes its cutoff value. However, with some properly chosen cutoff, *i.e.* 12, RSmatch could still achieve a comparable specificity with PatSearch. One possible explanation of getting high false positives for RSmatch could be that, with respect to the particular case of the HSL3 motif, its secondary structure conformation might be too pervasive in RNA sequences to be used as a discriminative feature. This could point out a problem concerning RSmatch's current scoring matrices, which need to be fine tuned to improve the tool's specificity. Good tuning could be achieved by setting up the scoring matrices through learning from a training data set. One interesting observation, however, was that RSmatch and PatSearch agreed perfectly upon the HSL3 locations for almost all of the true positives they found.

Table 5.2 Performance of RSmatch in the HSL3 experiment[a]

Cutoff Score	Selected Hits[b]	True Positives	Specificity	Sensitivity[c]
14	33	33	100.0%	53.2%
12	47	45	95.7%	72.6%
10	69	46	66.7%	74.2%
8	184	56	30.4%	90.3%

[a] PatSearch has a specificity of 98.2% and sensitivity of 87.1%;

[b] Hits whose scores are greater than or equal to the cutoff value used in this study are selected;

[c] Assume there are 62 mRNA structures containing the HSL3 motif, which include all histone mRNAs found by RSmatch and PatSearch.

Using the IRE motif, further comparisons between RSmatch and three other tools: PatSearch, stemloc and Rsearch, were performed. Default parameters for

Rsearch were adopted; for stemloc, the fold envelop was set to 1000. Instead of using the large UTR structure database described above, a small test date set was constructed to expedite the comparison process. First, PatSearch was used to search human UTR sequences for IRE motifs. Then for each hit sequence, its corresponding mRNA's $3'$ and $5'$ UTR sequence were selected. Following the same folding process as discussed before, these UTR sequences were folded to form the test data set. Totally, PatSearch found 27 hits, among which 9 were known true positives. Therefore PatSearch's specificity was 33%. These hits were from 23 distinct mRNA sequences. Assume that PatSearch had a 100% sensitivity. The $5'$ and $3'$ UTR sequences were extracted from the 23 distinct mRNAs and 46 UTR sequences were obtained. The 46 UTR sequences were then folded to get a small test data set, which contained 1196 structures. Using a known IRE-containing structure (NM_000032), which was one of the 9 true positives found by PatSearch, as the query, the small test data set was searched. Table 5.3 shows the results. Since Rsearch only accepts sequences, it was tested using only the primary sequence information in the test data set.

Except for the IRE-containing structure NM_001098, which was one of the 9 true positives found by PatSearch, and the query itself (NM_000032), all tools agreed on the IRE locations for the other seven true positives without salient discrepancy. Careful examination showed that NM_001098 was not properly folded to exhibit the existence of the IRE motif. RSmatch has the best specificity by ranking all seven true positives within its top 8 hits with only one false positive (NM_032484). Rsearch is close to RSmatch by ranking all seven true positives within its top 8 hits with one false positive (NM_003672). In contrast, stemloc gives five false positives within its top 10 hits. Setting different cutoff values yields different specificity and sensitivity for each tool. The point of balanced specificity and sensitivity appears at the cutoff value of 8 for all three tools. With this cutoff value, the specificity of RSmatch and Rsearch tied at $7/8 \times 100\% = 87.5\%$. This is better than the specificity of PatSearch

Table 5.3 IRE experiment results

True Positive	RefSeq ID	Location by PatSearch	RSmatch			Rsearch			stemloc		
			Location	Score	Rank	Location	Score	Rank	Location	Score	Rank
x	NM_000032[a]	13-35	-	-	-	-	-	-	-	-	-
x	NM_014585	203-229	202-231	21	1	202-231	34.11	1	202-226	13.021	5
x	NM_003234	3479-3511	3484-3506	19	2	3480-3510	31.42	2	3486-3503	15.936	2
x	NM_003234	3883-3913	3887-3909	17	3	3876-3925	27.80	6	3889-3906	10.914	7
x	NM_003234	3950-3976	3952-3974	17	3	3952-3974	25.50	10	3954-3971	10.476	9
x	NM_003234	3996-4024	3999-4021	17	3	3999-4022	28.53	4	4042-4048	1.149	25
x	NM_000146	19-41	20-40	16	6	7-51	27.84	5	17-41	8.574	12
	NM_032484	2353-2376	2358-2373	13	7	2355-2377	22.26	11	2354-2375	16.411	1
x	NM_003234	3429-3461	3434-3456	13	7	3433-3458	26.40	8	3436-3453	6.218	15
	NM_018992	2182-2205	2186-2202	12	9	2186-2202	18.43	19	2186-2202	11.459	6
	NM_003449	2160-2180	2163-2178	11	10	2160-2180	26.27	9	2161-2181	13.198	4
	NM_002081	3449-3469	3452-3467	11	10	3446-3472	20.47	17	3450-3470	8.290	14
	NM_173649	1371-1398	1431-1446	7	12	1372-1398	18.83	18	1376-1396	8.493	13
	NM_033337	1202-1226	1253-1257	5	13	1202-1227	21.52	14	1202-1227	4.540	18
	NM_001234	1106-1130	1157-1161	5	13	1106-1131	21.52	15	1106-11331	4.540	19
	NM_153706	174-194	108-119	5	13	171-198	17.49	20	219-234	4.400	20
	NM_003607	6892-6914	6851-6854	4	16	6892-6914	21.98	12	6930-6950	10.827	8
	NM_002086	82-102	126-129	4	16	94-117	16.48	22	101-125	2.833	22
	NM_012256	2594-2617	2571-2574	4	16	2536-2570	20.76	16	2590-2606	1.770	24
x	NM_001098	1-23	17-19	3	19	1-23	30.67	3	3-20	14.185	3
	NM_006731	4439-4465	4487-4489	3	19	4442-4462	21.65	13	4443-4460	9.920	10
	NM_003672	2556-2576	2592-2594	3	19	2547-2587	26.44	7	2558-2574	9.049	11
	NM_018234	2038-2058	2176-2178	3	19	2035-2061	14.11	24	2046-2058	4.986	16
	NM_024076	1799-1822	1816-1818	3	19	1800-1821	15.00	23	1832-1850	4.876	17
	NM_000877	3274-3294	3336-3338	3	19	3275-3293	13.81	25	3302-3321	3.182	21
	NM_003675	2-27	27-31	3	19	1-29	16.74	21	21-31	1.980	23
	NM_032323	1924-1944	1990-1992	3	19	1925-1943	12.43	26	1928-1948	0.678	26

[a] NM_000032 is used as the query structure for RSmatch, Rsearch, and stemloc. Thus there is no value (shown as "-").

(33%) and the specificity of stemloc (50%). The sensitivity of RSmatch, Rsearch and stemloc is 87.5%, 87.5% and 50% respectively. It is worth noting that RSmatch runs 30% faster than Rsearch; it took Rsearch 34 seconds to search the whole data set of 1196 structures while RSmatch used only 23 seconds. Consequently, RSmatch would be suitable for analyzing large data sets. It should also be pointed out that RSmatch permits wildcards in database searching and structure matching, which are not supported by Rsearch or stemloc.

5.2 Performance on Complex Structures

Further tests were conduct to measure how accurate RSmatch is for much complex structures. To this end, RNA structures and sequences were obtained from the Rfam database. 64 RNA families were used, each had more than 10 seed sequences and the consensus sequence length less than 250 nucleotides. For each RNA structure family, a structure was randomly selected as the query and searched against 10 randomly selected sequences belonging to the same family. To reflect real world scenarios, RNA sequences were extended at both $5'$ and $3'$ ends so that the length of a subject sequence was three times that of the original one. To ensure that the folded structures are long enough to fully contain the structure being investigated, it is required the moving window size be 1.5 times the length of the query RNA sequence. Furthermore, to include suboptimal structures, all structures with free energy within 1.5 kcal/mol above the minimum one were used. Compared with HSL3 and IRE, the 64 query structures were much more complex, with average length of 120nt and more than 70% of them comprised of nested loops and conjunctions.

To assess the accuracy, a measure called structure coverage denoted as p, was introduced and calculated by the following formula: $p = |Q_{align}|/max(|Q|, |S_{align}|)$, where $|Q_{align}|$ and $|S_{align}|$ are the lengths of aligned portion of query RNA and subject RNA, respectively, and $|Q|$ is the length of query RNA sequence. As shown in Fig.5.3, even though Rsearch has slightly more points clustered around high coverage area ($> 90\%$), the difference in the overall performance between RSmatch and Rsearch is not significant. In addition, the difference between RSmatch and Rsearch do not seem to be related to structure size or complexity. This result indicates that RSmatch has the ability to process complex structures.

5S rRNA family was then selected for further detailed tests. 5S rRNA has a length of 120nt, which contains several types of RNA structures, including hairpin, internal loop, bulge, and junction. There are 602 sequences in the 5S rRNA family,

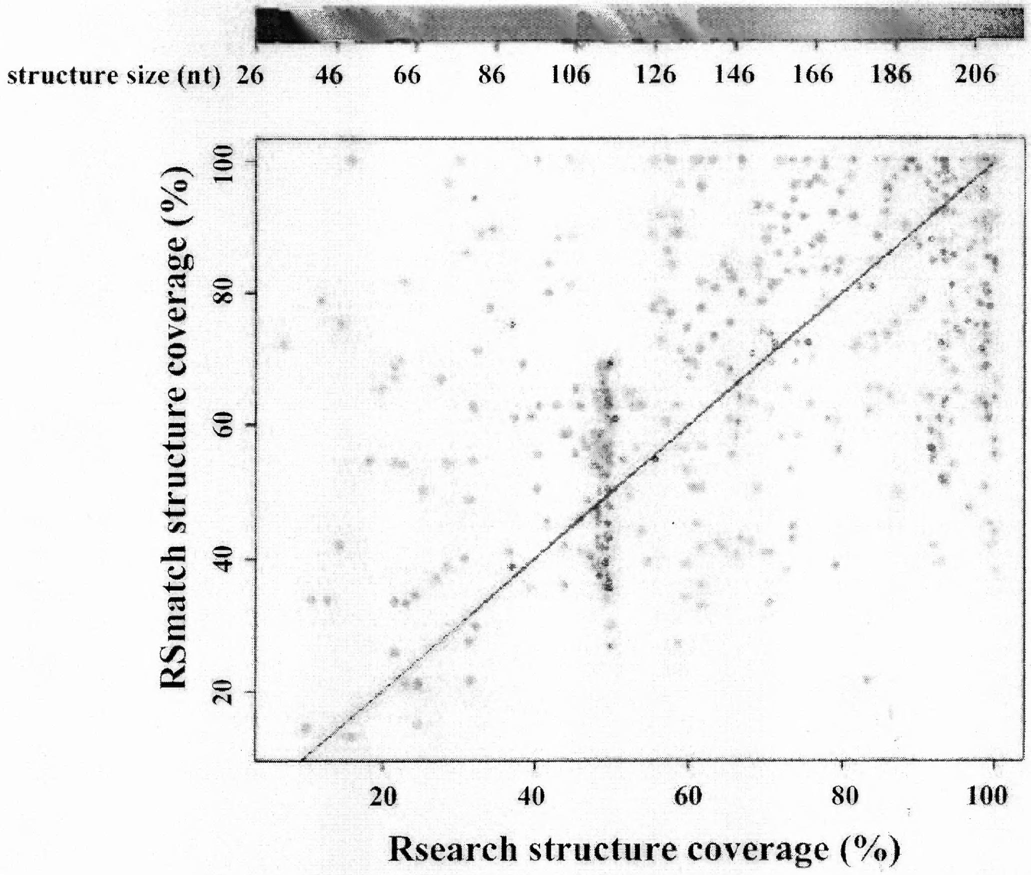


Figure 5.3 Performance comparison for 64 RNA families. Different colors are applied to represent structures of different sizes. Each point corresponds to one alignment between a query structure and a subject structure. The x-axis is the percent of coverage by Rsearch and y-axis is the percent of coverage by RSmatch.

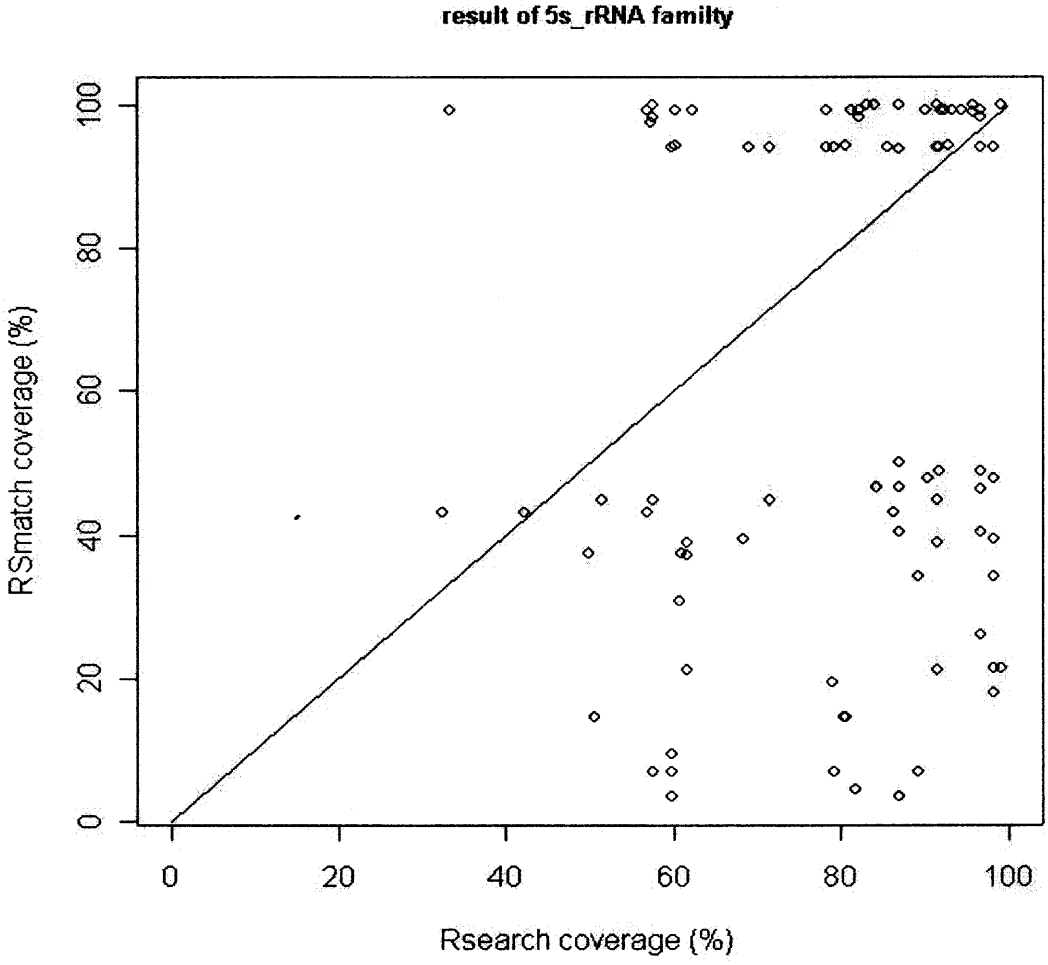
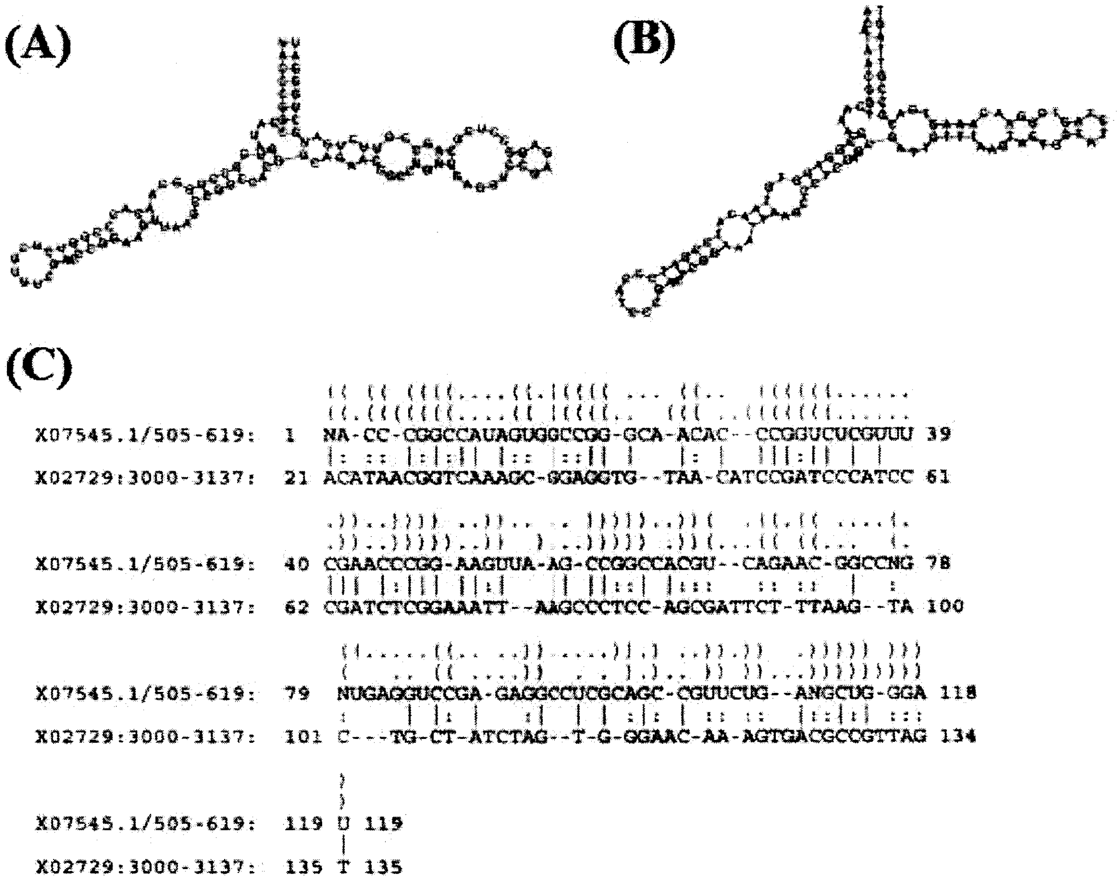


Figure 5.4 Performance comparison of 5S rRNA. A 5S rRNA was randomly chosen as the query structure and ten others as the subject sequences. The median value of the ten structure coverage values was then calculated. This process was repeated 100 times to generate 100 points for the graph. Therefore, each point represents one particular query structure.



```
                          (( (( ((((....((.((((( ... ((..  (((((((......
                          ((.((((((((....(( (((((..  ((( ..((((((((......
X07545.1/505-619:    1  NA-CC-CGGCCAUAGUGGCCGG-GCA-ACAC--CCGGUCUCGUUU 39
                          |: :: |:|:|| |:: ::|| |    |: |  |||:|| | |
X02729:3000-3137:   21  ACATAACGGTCAAAGC-GGAGGTG--TAA-CATCCGATCCCATCC 61

                          .))..)))) ..))..  . ))))).)) (  .((.(( ....(.
                          .))..))))))..))  ) ..)))))) .))(...(( ((...  (.
X07545.1/505-619:   40  CGAACCCGG-AAGUUA-AG-CCGGCCACGU--CAGAAC-GGCCNG 78
                          ||| |:||| ||:|    | ||::| |::    :: ::  |  :
X02729:3000-3137:   62  CGATCTCGGAAATT--AAGCCCTCC-AGCGATTCT-TTAAG--TA 100

                          ((.....((.. ..)) ....)).) ..)).))  .))))) )))
                          (    .. (( ....))  . . ).).. )) ))...))))))))))
X07545.1/505-619:   79  NUGAGGUCCGA-GAGGCCUCGCAGC-CGUUCUG--ANGCUG-GGA 118
                          :      | |: |    :|  | | :|: | :: ::  |::|:| :::
X02729:3000-3137:  101  C---TG-CT-ATCTAG--T-G-GGAAC-AA-AGTGACGCCGTTAG 134

                          )
                          )
X07545.1/505-619:  119  U 119
                          |
X02729:3000-3137:  135  T 135
```

Figure 5.5 An example alignment of two 5S rRNA was shown: the query structure is X07545/505-619, the subject RNA is X02729; and the detailed alignment by RSmatch.

suitable to carry out a thorough analysis. One 5S rRNA was randomly chosen as query structure and ten others as subject sequences for alignment. This process was repeated 100 times. The performance comparison of Rsearch and RSmatch is shown in Fig.5.4. For 5S rRNA, RSmatch outperforms Rsearch in discovering the overall sequence more completely. An exemplary alignment is shown in Fig.5.5.

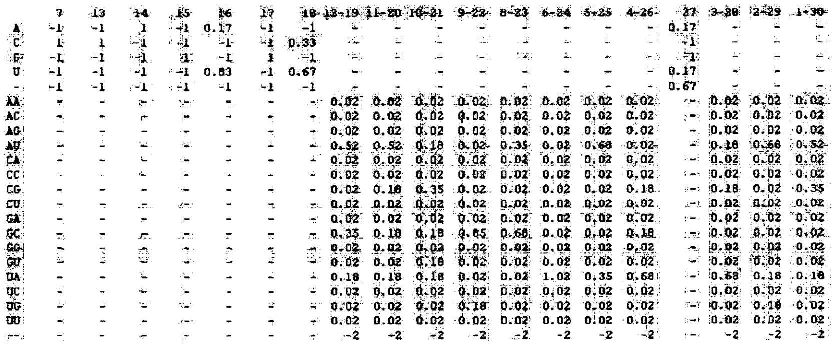
5.3 Multiple Structure Alignment and Iterative Database Search

RSmatch algorithm has been extended to conduct multiple structure alignment. An example using IRE is shown in Figure 5.6. While the alignment algorithm is the same, the multiple alignment function uses a position-specific scoring matrix (PSSM, Figure 5.7). For a given set of structures, the multiple alignment function first identifies

```
# STOCKHOLM 1.0

NM_000032:1-52                  TTCGTTCGTCCTCAGTGCAGGGCAACAGGA
NM_014585:151-250               CAACTTCAGCTACAGTGTTAGCTAAGTTTG
NM_003234:3944-4043             AATTATCGGGAACAGTGTTTCCCATA-ATT
NM_003234:3844-3943             CATTATCGGGAGCAGTGTCTTCCATA-ATG
NM_003234:3444-3543             AATTATCGGAAGCAGTGCCTTCCATA-ATT
NM_003234:3394-3493             ATTTATCAGTGACAGAGTTCACTATA-AAT
#=GC SS_cons                    (((((.(((((......).))))))).)))
//
```

Figure 5.6 Multiple structure alignment of several IRE structures.



	7	13	14	15	16	17	18	12-19	11-20	10-21	9-22	8-23	6-24	5-25	4-26	27	3-28	2-29	1-30
A	-1	-1	1	-1	0.17	-1	-1	-	-	-	-	-	-	-	-	0.17	-	-	-
C	1	1	-1	-1	-1	-1	0.33	-	-	-	-	-	-	-	-	-1	-	-	-
G	-1	-1	-1	1	-1	1	-1	-	-	-	-	-	-	-	-	-1	-	-	-
U	-1	-1	-1	-1	0.83	-1	0.67	-	-	-	-	-	-	-	-	0.17	-	-	-
-	-1	-1	-1	-1	-1	-1	-1	-	-	-	-	-	-	-	-	0.67	-	-	-
AA	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
AC	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
AG	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
AU	-	-	-	-	-	-	-	0.52	0.52	0.18	0.02	0.35	0.02	0.68	0.02	-	0.18	0.68	0.52
CA	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
CC	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
CG	-	-	-	-	-	-	-	0.02	0.18	0.35	0.02	0.02	0.02	0.02	0.18	-	0.18	0.02	0.35
CU	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
GA	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
GC	-	-	-	-	-	-	-	0.35	0.18	0.18	0.85	0.68	0.02	0.02	0.18	-	0.02	0.02	0.02
GG	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
GU	-	-	-	-	-	-	-	0.02	0.02	0.18	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
UA	-	-	-	-	-	-	-	0.18	0.18	0.18	0.02	0.02	1.03	0.35	0.68	-	0.68	0.18	0.18
UC	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
UG	-	-	-	-	-	-	-	0.02	0.02	0.02	0.18	0.02	0.02	0.02	0.02	-	0.02	0.18	0.02
UU	-	-	-	-	-	-	-	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	-	0.02	0.02	0.02
--	-	-	-	-	-	-	-	-2	-2	-2	-2	-2	-2	-2	-2	-	-2	-2	-2

Figure 5.7 PSSM of the multiple alignment of IRE in Fig.5.6. Each column in the PSSM corresponds to the position of a structure component, either single base or base pair. Position of a single base is represented by the nucleotide number and position of a base pair is represented by two nucleotide numbers connected by a dash. For each column, the scores of individual structure components in that position are listed in rows where "-" means not applicable.

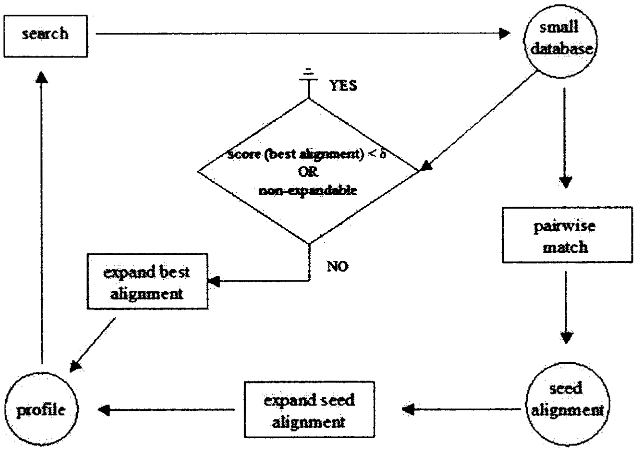


Figure 5.8 Flowchart of multiple structure alignment.

the best alignment of two structures, and builds a PSSM. The PSSM is then used to search for the closest structure in the rest of the set. A flowchart of multiple structure alignment is shown in Figure 5.8. If the alignment score of a structure to the PSSM is above a cutoff (user-defined), it is selected and its structure is used to update the PSSM. This step is iteratively conducted until no structures have alignment score above the cutoff. In a sense, this method employs an implicit guided hierarchical tree using the average value for joining nodes. As an example, from human UTR database 6 IRE-containing structures were selected and other 6 none-IRE structures were randomly chosen to form a small dataset and RSmatch was run against it. The output is shown in Figure 5.6. The final result is in Stockholm format for multiple structure alignment. Conceivably, when the given set of structures is a large database, the multiple structure alignment function of RSmatch in effect conducts iterative search for finding similar structures.

5.4 Discussion and Conclusions

The work presented here is intended to provide an efficient tool to directly perform structure alignment and search of RNA secondary structure databases. Its capability to carry out multiple structure alignment and iterative database search can potentially be used to uncover RNA motifs *ab initio.* For example, one can use an RNA structure of interest to search an RNA structure database, and build PSSM iteratively to build an RNA motif, as demonstrated for IRE in this study.

RSmatch bears similarities to rna_align and RNAforester in that the structural particularities are either explicitly captured using hierarchical tree/forest structures or implicitly represented using arc-annotated structures. However, RSmatch differs from rna_align and RNAforester in two major aspects. First, RSmatch keeps structural consistence by only allowing single bases matched with single bases and base pairs matched with base pairs whereas rna_align and RNAforest don't impose this restriction. Second, RSmatch keeps the integrity of single-stranded regions by matching one with another, instead of breaking a single-stranded region into pieces and aligning them with different single-stranded regions. In addition, RSmatch has less time and space complexities than the other two tools.

The concept of circles introduced in this paper is reminiscent of the "k-loop" described in the classic RNA structure prediction paper. The difference is that the circles can reflect the inter-base-pair relationship by focusing on two base-pairs at a time while the "k-loop" cannot. By organizing all circles into a hierarchy tree, the overall structural particularity can be captured. It should also be pointed that there is a major difference between the hierarchy tree introduced here and the parse trees of SCFG. The hierarchy tree is constructed from circles and aims to obtain the panorama of the secondary structure of RNA at a higher level than that of the SCFG parse tree, while detailed information is still available within each circle in the tree. With the introduction of partial structures, this two-level structure modeling (intra- and inter- circles) makes it possible to develop an efficient algorithm that runs at time O(mn) as shown in this thesis.

The proposed approach takes full advantage of structure prediction techniques. It separates RNA folding from structure alignment. Simultaneous RNA folding and alignment is believed to be the optimal solution for both finding the right structure and locating homologous sub-structures of RNAs. Unfortunately, it is computationally prohibitive for even a moderate number of RNAs. Some improvements have been proposed, but extensive computing time still makes them infeasible for database searches. By separating the process into two steps, the computing efficiency has been greatly enhanced, making it possible to process a large-scale pre-folded RNA structure database for homologous motifs. However, a drawback of using pre-folded RNAs is that the prediction tools may not produce correct RNA structures, as observed in the experiments. It is estimated that the RNA folding programs solely based on thermodynamic properties of RNA can correctly predict RNA structures with about 70% of chance. Secondly, higher complex structures, such as pseudoknots, cannot be predicted in most commonly used programs, including the Vienna RNA package used in this study. A solution to removing the first drawback is to choose suboptimal structures in addition to the optimal one to increase the chance of obtaining correct structures. It has been reported that using suboptimal structures whose thermodynamic free energies are within 2% of that of the optimal one can greatly improve the structure prediction of RNA. In the IRE experiments shown above, it was found that the predicted structure for NM_001098/1-23 did not exhibit the existence of an IRE motif. By relaxing the free energy range, the IRE motif was finally detected from one suboptimal structure whose free energy was 1.7kcal/mol higher than the optimal one. Because of the computing efficiency of the program, an increase of the number of RNA structures does not impose big burden on database searching (data not shown). The cost will be at the database building stage, which is however done only once.

The moving window approach used to extract and fold subsequences was aimed to make the folding process more accurate and efficient. This is because RNA folding programs are known to have pronounced difficulties in correctly predicting large RNA structures. Furthermore, predicting the structure for a long sequence takes much longer time than predicting structures for its subsequences. Another advantage of using the moving window method is that small motifs falling in the overlapped subsequences could be folded twice, increasing the chance of their being detected.

Pattern-based tools, such as PatSearch and RNAmotif, use descriptions of an RNA structure as queries to search a sequence database for similar structures. This type of search does not take into consideration the context of a hit sequence, which could influence the (sub)structure of the sequence. For example, as shown in the experimental results, PatSearch can achieve a satisfactorily high specificity when the structure of a pattern is not flexible and its description is relatively precise, such as the HSL3 motif. However, the sensitivity of PatSearch is low with rigid pattern descriptions. For relatively flexible structures, such as IREs, the specificity of PatSearch drops because it does not take into account the context in which a motif is located. On the other hand, using folded RNA structures, the proposed RSmatch tool overcomes these shortcomings with a high specificity, thus complementing the pattern-based tool. However, as also shown in the experimental results, the error existing in folding an RNA sequence (NM_001098) can lower the sensitivity of RSmatch. It may suspect that the inaccuracy introduced by RNA folding could be a bottleneck for the proposed technique in achieving a very high sensitivity.

The scoring matrices for single-stranded and double-stranded regions and the gap penalty assignment are very primitive in the sense that they are not based on any probabilistic model or learned from any training data set. One interesting observation in the HSL3 experiment was that RSmatch did find most HSL3 sites correctly. However, the scoring scheme seemed not acute enough to filter out many false positives. Part of the problem is that there are not enough motifs that can be used to construct optimal scoring matrices. In fact, tests were conducted using the matrices (RIBOSUM) proposed by Klein and Eddy, which were built upon small subunit ribosomal RNAs. No discernible difference was found in the HSL3 experiment, in which both matrices were used (data not shown). Another related question is whether different types of RNA, such as tRNA, rRNA, and UTRs, need their own scoring matrices. It is conceivable that large highly structured RNAs, such as rRNA, may be able to tolerate more mutations than short RNA motifs that occur in UTR regions. If so, using different scoring matrices for different types of RNAs will be warranted. Furthermore, it is possible that the mutation rate is different for nucleotides in different regions of an RNA motif. Therefore, PSSM might be more suitable in these cases. To this end, the iterative search function of RSmatch, which searches a database using PSSM, can be applied.

Motivated by the statistical methods of assessing results in sequence alignment, attempts were tried to develop scores of the database search with known probabilistic distributions. The score distribution seemed close to be normal (data is not shown). However since the scoring scheme is still at its preliminary stage and much is to be learned about the RNA structure database presented in the paper, search results are presented in terms of ranking. More elaborate statistical assessment of the search results will be developed in the future.

CHAPTER 6

MINING CONSERVED RNA STEM-LOOPS IN HUMAN AND MOUSE UTRS

UnTranslated Regions (UTRs) of mRNAs constitute a large proportion of the gene-coding sequence in mammalian genomes. UTRs are involved in various steps of mRNA metabolism, including RNA localization, translation, and stability. Regulation of gene expression through UTRs occurs in diverse cellular pathways and at various developmental stages. Several RNA stem-loop structures in UTRs have been experimentally identified, including the histone $3'$-UTR stem-loop structure (HSL3) and iron response element (IRE). These stem-loop structures are conserved among mammalian orthologs, and exist in several genes with similar functions. It is not known, however, to what extent stem-loop structures like these exist in human UTRs.

This chapter took a systematic approach to mine stem-loop structures in human and mouse UTRs. Special interest was on RNA stem-loops that were conserved between human and mouse orthologs and existed in genes with similar functions or involved in the same pathway. By comparing RNA structures between human and mouse orthologous genes, and then among all human genes, followed by combining Gene Ontology information, 30 RNA stem-loop structure groups were identified. The result indicates that there exist more conserved stem-loop structures in UTRs, but their conservations are less than those of HSL3 and IRE.

6.1 Introduction

Post-transcriptional control is one of the mechanisms that regulate gene expression in human cells. RNA elements residing in the UnTranslated Regions (UTRs) of mRNAs have been shown to play various roles in post-transcriptional control, including mRNA localization, translation, and mRNA stability [17, 18, 44]. RNA elements in UTRs can be roughly divided into two groups: elements whose functions are primarily attributable to their sequences and elements whose functions are attributable to their secondary or tertiary structures. For simplicity, they are called sequence elements and structure elements respectively. Well-known sequence elements include AU-rich elements (ARE), which contain one or several tandem AUUUA sequences and are involved in RNA stability [62–64], and miRNA target sequences, which have sequence partially complementary to cognate miRNA sequence and involved in translation or stability [65].

Among all structural elements, the histone 3′-UTR stem-loop structure (HSL3) and the iron response element (IRE) have been most extensively studied [37, 38]. In either case, both sequence and structure are important for the functions of the structural elements. HSL3 is a stem-loop structure of 25 nt that exists in 3′-UTRs of most histone genes. The structure is critical for both the termination of transcription of these histone genes, and the stability of histone mRNAs. HSL3 structures bind the stem-loop binding proteins (SLBP). IRE is a stem-loop structure of 30 nt with a bulge or a small internal loop in the stem. IREs have been found in both 5′-UTRs and 3′-UTRs of mRNAs whose products are involved in iron homeostasis in higher eukaryotic species. IREs bind the iron regulatory proteins (IRPs), which control translation and stability of IRE-containing mRNAs. HSL3 and IRE are similar in several aspects: both are small simple RNA structures less than 40 nt; both exist in UTRs of several genes with related functions; and both bind cellular protein and exert post-transcriptional gene regulation. The regulations via HSL3 and IRE constitute a distinct mode of gene regulation, whereby several genes can be regulated via a common RNA structure in UTRs. It is not known, however, to what extent human mRNAs are subject to the same mechanism.

Identification of functional sequence motifs in genomes has been heavily studied in recently years, particularly for the promoter region and sequences related to splicing

[66–71]. However, RNA structure elements have been investigated to a much lesser extent, largely due to the difficulties in predicting right RNA structures and conducting RNA alignments and huge computing costs involved. While some successes have been achieved using phylogenetic approaches to gain accuracy in RNA structure prediction [72–74], large-scale mining of conservative structures in human UTRs has not been attempted.

Here a systematic approach is presented to uncover novel conserved RNA stem-loops in human and mouse UTRs. Using the newly developed RNA structure alignment tool RSmatch, the first step is to compare RNA structures in UTRs from human and mouse orthologs. Using cluster analysis and Gene Ontology information, RNA structures that existed in more than 2 genes that share common functions, and are involved in the same biological pathways, were identified. Finally, cross-validated RNA structures were identified from human UTRs by mouse UTRs. Overall, 30 RNA stem-loop structure groups were detected, including HSL3 and IRE. This bioinformatic study lays a ground work for future wet lab validation of putative conserved RNA stem-loops in human and mouse UTRs, and represents a framework which can be used to discover RNA structural elements in other studies.

6.2 Materials and Methods

6.2.1 UTR Sequence and Structure Databases

From National Center for Biotechnology Information (NCBI), 28,926 human and 26,243 mouse RefSeq mRNA sequences (January '04 versions) were downloaded. The information regarding human and mouse orthologs was obtained from the HomoloGene database (`ftp://ftp.ncbi.nih.gov/pub/HomoloGene/`). UTRs of RefSeq sequences were extracted according to RefSeq's GenBank annotation. RNA structures in UTRs were prepared by a method called "slide and fold", as described in [75]. Briefly, for each UTR sequence, 100nt subsequences were taken at every 50nt nucleotide position from 5' to 3', making consecutive subsequences overlap with one another on a 50nt segment. Subsequences shorter than 100nt, e.g. at the 3' end, were also kept. Then all subsequences were folded using the RNAsubopt function in the Vienna RNA package [39, 40], with the setting "-e 0". With this setting, multiple structures with the same minimum energy can be generated. Using the method, 575, 410 structures from human UTRs, and 445, 106 structures from mouse UTRs were obtained.

6.2.2 RNA Structure Comparison

Pair-wise comparisons of RNA structures were carried out by RSmatch [75], with the "dsearch" function and default scoring matrices for single-stranded (ss) and double-stranded (ds) regions. Specifically, nucleotide match scores are 1 and 3 in ss and ds regions, respectively; and mismatch scores are -1, and 1 in ss and ds regions, respectively. Gap penalty is -6 for both ss and ds regions. This scoring scheme in effect favors structure matches. The focus was on on three values from each comparison: sequence length of the alignment, size of the ds region of the alignment, and score from RSmatch.

6.2.3 Randomization of UTR Sequences

Randomization of UTR sequences was carried out by PERL using a 5-order Markov Chain Model. Briefly, 1, 000 human and mouse orthologous gene pairs were randomly chosen. For each sequence, the occurrences of all 6 and 5 nucleotides (hexamers and pentamers) were calculated and used to derive transition probabilities for the MC model. The MC model was then used to generate randomized UTR sequences, which were processed by the same method for real UTR sequences.

6.2.4 Comparison of RNA Structures Among All Genes

Human RNA structures selected from the comparison of human and mouse orthologs were further compared among genes. Structures that were similar to other structures from at least 3 distinct genes with the similar score ≥ 17 were selected. Overall, 2, 054 out of $6,345$ structures met the criteria. To assess the false positive rate of this step, randomization on all human RNA structures (6,345) was conducted by randomly swapping nucleotides in the ss and ds regions, respectively, while maintaining the overall structure. Randomized structures were treated as real structures.

6.2.5 Cluster Analysis of RNA Structures

To cluster RNA structures, the normalized dissimilarity scores $D_{i,j}$ between all structures were first calculated; $D_{i,j} = (S_{max} - S_{i,j})/S_{max}$, where $S_{i,j}$ was the similarity score derived from RSmatch using the local structure alignment function between structures i and j, and S_{max} was the maximum similarity score of all structure comparisons. For cluster analysis, the hierarchical clustering function in R with the "average linkage" method was used for joining nodes. To select groups of RNA structures, the "cutree" function was applied to cut the hierarchical tree into groups using the normalized dissimilarity scores, which were also called heights in the tree. Structures in each group were aligned by the multiple structural alignment function of RSmatch with default scoring matrices. Structures in the same group were also compared pair-wise; the average of all pair-wise similarity scores for the group was called Cohesive Value (CV), which indicated the degree of structural closeness the member structures in the group are to each other.

To assess the quality of structure groups, the following method was applied to derive expected CVs for groups: from the 2054 motif structures, the method randomly selected a defined number of structures and calculated the CV. This process was carried out for groups with 4-20 structures. For each group, the process was repeated 100 times. The average of all CVs for a particular group size is its expected value. Groups with CV above the corresponding expected value were selected for further analysis.

6.2.6 Gene Ontology Analysis

Three Gene Ontology (GO) categories, *i.e.* molecular function (MF), biological process (BP) and cell component (CC), were downloaded from Gene Ontology consortium (`http://www.geneontology.org/`). The mapping between genes and GO entries was obtained from LocusLink database [76]. Hypergeometric analysis was used to assess whether an RNA structure group was significantly associated with some GO entries, as previously described [77]. Structure groups with p-value < 0.05 were selected.

6.2.7 Cross-validation with Mouse UTR Structures

Selected human RNA structure groups after GO analysis were compared to their orthologous mouse structures. For each group, mouse UTR structures corresponding to human structures in the group were retrieved and compared by the multiple structure alignment function of RSmatch. For each group, the consensus structure of human RNA structures was compared to its mouse counterpart. An RNA structure group was selected if (1) the human consensus was identical to the mouse one, or (2) the human consensus was contained within mouse one or *vice verse.* In the latter case, a consensus of human and mouse structures was built to represent the structure.

6.3 Results and Discussion

In this study, the final goal is to identify stem-loop structures in human and mouse UTRs that may play roles in post-transcriptional gene regulation. Particular interest is in finding structures with properties like those of HSL3 and IRE: (1) They are located in UTRs; (2) they contain stem-loop structures with a size 30nt; (3) They are conserved between human and mouse orthologs; (4) They occur in several genes that have similar functions or are involved in the same biological pathways; To this end, a mining strategy was developed and depicted in Fig.6.1.

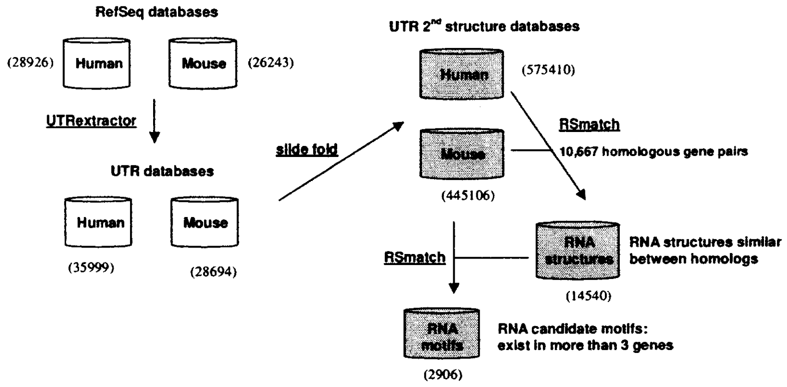


Figure 6.1 Flowchart of mining conserved RNA stem-loops in human and mouse UTRs. The number of entries for each database, either sequences or structures, is shown in parenthesis. A "slide and fold" method (see Materials and Methods for details) was used to fold subsequences of human and mouse UTRs. Structures from human and mouse orthologous genes were compared using RSmatch and the aligned substructures were kept as "conserved RNA structures".

First, UTR sequences were extracted from RefSeq sequences obtained from NCBI. Then a "slide and fold" method was then adopted to construct RNA structures in UTRs (see Materials and Methods for details). With this method, subsequences in UTRs, 100nt long or less, were folded using the Vienna RNA package. Adjacent subsequences overlapped by 50nt. This method can derive RNA structures accurately and efficiently for two reasons: (1) Predicting small structures is more accurate and efficient than big ones; (2) Structures with the size less than 50nt were folded twice, further increasing the chance of getting accurate RNA structures. These two advantages are particularly relevant for mining small RNA structures in this study. In addition, the setting used in the Vienna package could give rise to multiple RNA structures with the same minimum energy for a given sequence and thus further improved the folding accuracy. This step resulted in 575, 410 RNA structures from human UTRs and 445, 106 RNA structures from mouse UTRs.

RNA structures from human and mouse orthologs were compared. For each orthologous gene pair, the set of RNA structures from the human gene were compared with the set of structures from the mouse gene. Alignments with a positive score were kept. In order to assess the significance of the alignments, three values of a structure alignment were considered: size of the alignment, size of the ds region of the alignment, and RSmatch score of the alignment. The goal was to obtain expected values from randomized structures to select real aligned structures. To this end, sequences were randomized using a 5-order Markov Chain (MC) Model. The reason to choose 5-order MC model is because sequence elements that are 6nt or shorter still remain after randomization, thereby separating sequence conservation from structure conservation. For each aforementioned value type, the expected value was the 95th percentile of all values from the randomized set. For the size of aligned structure, the size of ds region, and the RSmatch score, the expected values were 23nt, 14nt, and 17, respectively (Figures 6.2, 6.3, 6.4). For each histogram, the 95th-percentile value is indicated by a vertical dotted line, *i.e.* 23 in 6.2, 14 in 6.3, and 17 in 6.4. Structure alignments that had at least two of three values higher than the respective expected values were selected. In addition, structure alignments in which two comparing structures were identical at the sequence level were eliminated. This step resulted in 6,345 alignments.

To make the proposed approach computationally efficient, the focus was on human RNA structures. Since special interest was on structures that existed in several genes with related functions, an all-against-all pair-wise comparison of all $6,345$ RNA structures was conducted. Each comparison gave rise to an alignment score. Structures that were similar to at least three other structures with the alignment score ≥ 17 were selected. $2,054$ RNA structures were obtained at this step (Fig. 6.5). To assess the false positive rate of the selection criteria, RNA structures were randomized by swapping nucleotides in both ss and ds regions, while keeping the

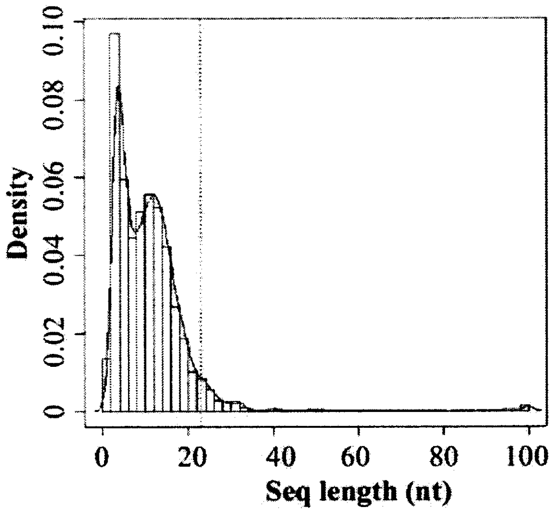


Figure 6.2 Histogram of the sequence length of the aligned substructures.

overall secondary structure intact. With the same selection criteria, 851 structures from the randomized set could be selected. Thus, 60% (1,203/2,054) structures as selected were significant.

To group similar RNA structures, hierarchical clustering was adopted. First, using pair-wise alignment scores, the normalized dissimilarity scores can be derived to represent distances among the structures. It was straightforward then to construct a hierarchical tree containing all 2,054 structures based on their mutual dissimilarities (Fig.6.6(a)). Hierarchical tree can be cut at different heights to give rise to subtrees representing groups. To group structures, all possible heights were first obtained to form a distribution of the number of groups (Fig.6.6(b)). Then, 100 distinct heights were selected at every percentile of the distribution. Using these 100 values to cut the tree, totally 58,247 groups of structures were obtained, each containing several structures.

Since special interest was on structures that existed in multiple genes with similar functions, RNA groups were further studied by their Gene Ontology (GO) information. The hypergeometric test was adopted to measure the significance of the

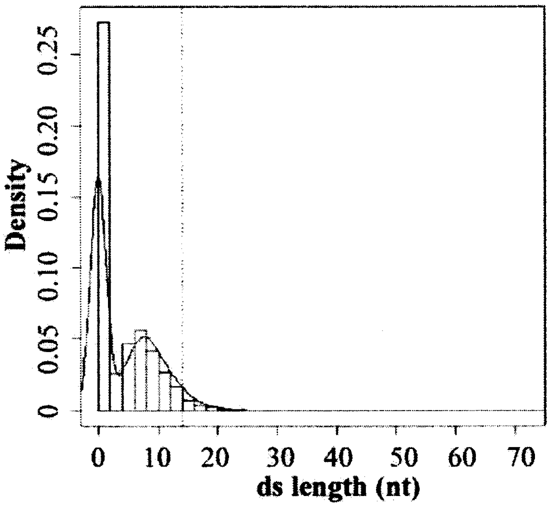


Figure 6.3 Histogram of sequence length in the ds region.

association of a structure group and a particular GO entry. If a structure group was significantly associated with a GO entry (p-value ≤ 0.05), and more than one structure in the group was associated the GO entry, the group was considered significantly related to the GO entry, and selected for further analysis.

To measure how similar member structures in each selected group are to one another, a measurement called Cohesive Value (CV) was introduced, which was the average of all pair-wise similarity scores among structures in the same group. To assess how significant a group was with respect to the similarity among structures in the group, the same number of structures from 2,054 structures were randomly chosen to form a group, and its CV was calculated. For a given group size, the process was repeated 100 times and the mean value was used as the expected CV for groups of the same size. Since the number of structures in a group ranged from 4–20, expected values for groups with 4–20 structures were derived in Fig.6.7. 1323 structure groups were obtained with CVs above the corresponding expected values. Overall, 1551 structures were selected. In addition, since one structure may exist in several groups due to different heights used in cutting the hierarchical tree, groups that overlapped

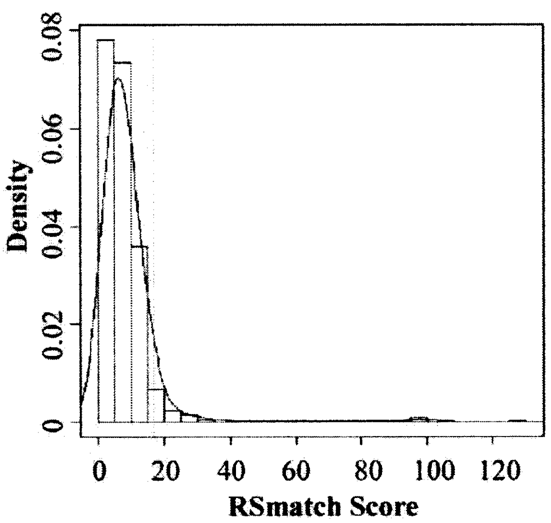


Figure 6.4 Histogram of RSmatch scores.

with other groups with better CVs after the GO analysis were eliminated. This step resulted in 1051 groups of structures, corresponding to 1399 RNA structures.

Finally, the cross-validation was conducted on the human structure groups by their mouse orthologs. For structures in each structure group, their corresponding mouse structures were retrieved (Fig.6.1). Mouse structures were aligned by the multiple structure alignment function of RSmatch to give rise to a consensus structure. If the consensus structure from a human group was the same as that from its corresponding mouse group, or one was part of the other, the structure group was then considered as validated. This step resulted in 30 groups of RNA structure groups corresponding to 370 structures.

6.3.1 Identified RNA Structure Groups

Among 30 groups of RNA structures, HSL3 and IRE ranked the 1st and 2nd with respect to CV. This result not only validated the proposed approach, but also indicated that other groups of RNA structures also existed, but probably not as well conserved as HSL3 or IRE. Using the multiple alignment function of RSmatch, consensus structure was generated for each structure group. In a sense, each structure group

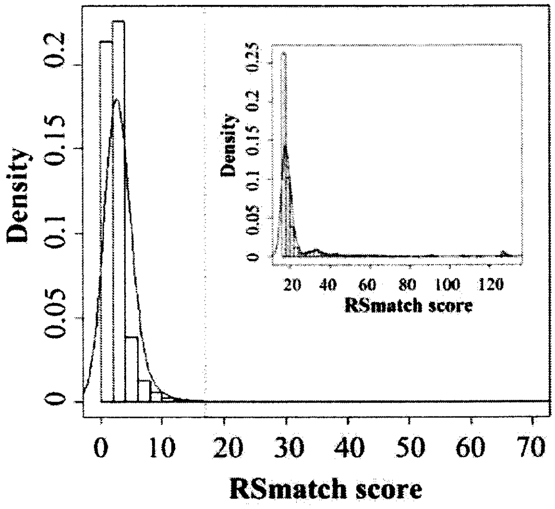
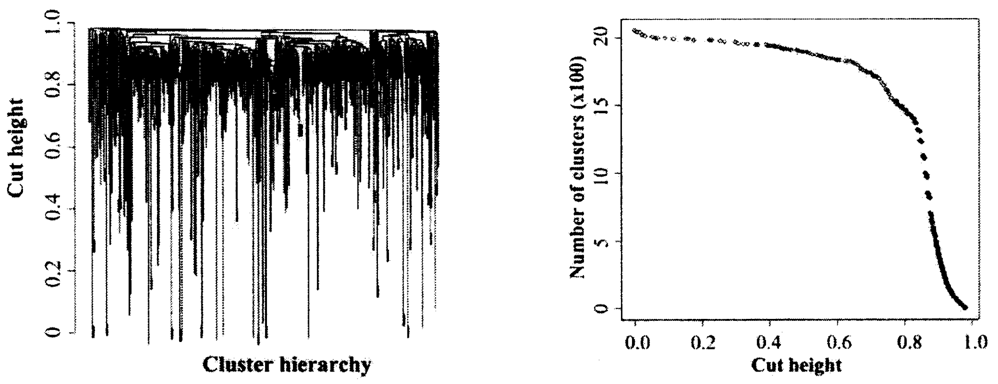


Figure 6.5 Histogram of RSmatch scores derived from pair-wise comparisons of 6,345 human RNA structures. Values above 17 were re-plotted as shown inside the main plot.

represented a putative RNA structural element. The structure sizes of the consensus structures ranged from 15 to 31.

While various randomization methods were incorporated in the process with the goal to keep false positive rate low, spurious structure groups can still exist. The exact false positive rate is hard to calculate as multiple steps are involved. An estimate is 60% based on the assessment during the all-to-all pair-wise alignment step. False negatives can also arise at several steps. First, it is known that RNA structure prediction is not 100% accurate. This would affect RNA structure groups with smaller number of structures more. Second, in order to separate conserved structures from conserved sequences, structures that were identical between human and mouse orthologs at the sequence level were eliminated. This step could affect RNA structures that are perfectly conserved between human and mouse. Third, some structures may reside in genes that are not functionally related, or GO information for their genes is not yet available.



(a) Hierarchical clustering

(b) Cluster distribution

Figure 6.6 Cluster analysis result.

Despite false positives and false negatives, some RNA structure groups look promising and warrant further wet lab validations. Since it is generally believed that for stem-loops, nucleotides in the loop region are more likely to participate in the nucleotide-specific recognition by RNA binding proteins, structure groups with conserved nucleotides in their loops will be of greater interest for validations. For example, among the final 30 groups, there was one structure group related to transcription and has a step-loop structure with a conserved `G` in the loop; and several structures related to cell proliferation, cell cycle, and cell growth and/or maintenance have the `CAGA` sequence in the loop.

6.4 Conclusion

A systematic approach was proposed to mine stem-loop structures in human and mouse UTRs. This approach involved comparing RNA structures between orthologous genes, and among all genes, analyzing Gene Ontology information, and cross-validation of human and mouse structures. The final result identified 30 RNA stem-loop structure groups corresponding to 370 structures. This result indicates that there exist more conserved stem-loop structures in UTRs, but their conservations are less than those

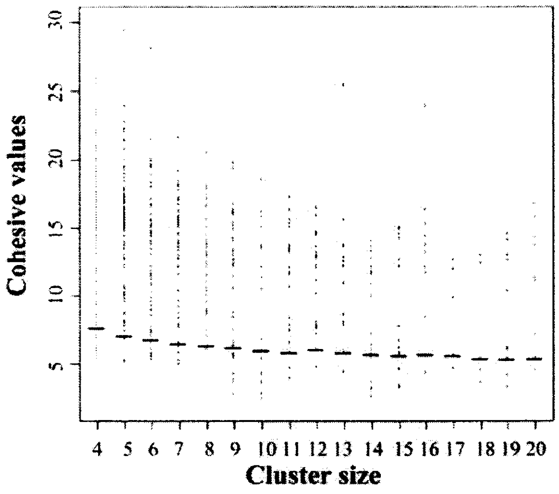


Figure 6.7 Structural cohesiveness of all clusters from the tree in Fig.6.6(a) cut with different heights.

of HSL3 and IRE. This bioinformatic study lays a ground work for future wet lab validation of putative conserved RNA stem-loops in human and mouse UTRs, and represents a framework which can be used to discover RNA structural elements in other studies.

CHAPTER 7

IMPLEMENTED SOFTWARES AND ONLINE SERVERS

7.1 RSmatch Software Package

The most important software package is RSmatch. The package described takes into account both sequence and structural information of RNA, allowing alignment of structure-annotated RNA sequences. The package can be applied to RNA motif discovery and database search. In the current alpha version 1.0, RSmatch provides four functions: (1) regular database search, (2) multiple structure alignment, (3) iterative database search, and (4) pairwise sequence alignment.

1. For a regular database search, the package finds RNA structures in a database that locally or globally match a given (usually small) query structure. The found local regions may become candidate sites containing an RNA motif. This function can also be used to detect motif occurrences in an RNA structure when the query structure is a known motif.

2. The function of multiple structure alignment constructs a multiple local alignment for a given set of RNA structures, by progressively expanding the alignment at each stage. This is a useful tool when a small set of RNAs are functionally related by a shared motif. This shared motif could be located by the multiple local alignment function.

3. Iterative database search is an extension of the regular database search. It is able to restart a new round of database search and update the employed scoring scheme based on the latest retrieval result. This function could be much more sensitive but also much slower than the regular database search.

4. The function of pairwise sequence comparison requires installation of the Vienna RNA package. Using a sliding window method, this function folds two given

RNA sequences and detects the region(s) that could be treated as the putative motif(s) shared by the two RNA sequences.

In RSmatch, two scoring schemes, referred to as position independent and position dependent schemes, are implemented. The position independent scheme consists of two scoring matrices, one for single-stranded and the other for double-stranded regions. This scoring scheme is used in the regular database search and pairwise sequence comparison functions. The position dependent scheme, also known as a profile, scores individual structure positions and is used by the multiple structure alignment and iterative database search functions. RSmatch provides both global and local alignment options even though the latter is of more interest.

Compared with other tools for RNA structure alignment, RSmatch is faster, requiring quadratic time in the size of two given structures.

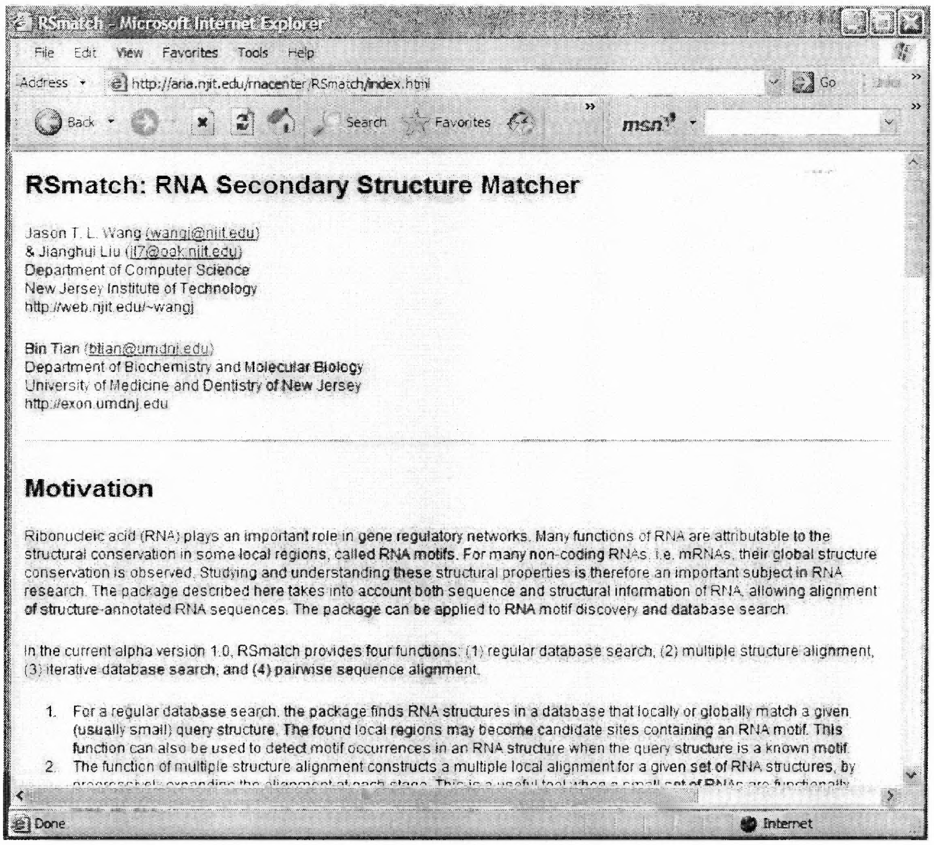
7.1.1 Download

The alpha version 1.0 of RSmatch can be downloaded from `http://aria.njit.edu/rnacenter/RSmatch/` (ref Fig.7.1). This is a relatively stable version. Effort are continuously made to improve the package. To obtain newer, perhaps unstable, versions, send email to the authors.

7.1.2 Installation

The RSmatch package is implemented using Java and Perl and run under a UNIX operating system. It needs a Java environment to run smoothly. If the input data are RNA sequences (which must be in the FASTA format), it is needed to download and install the Vienna RNA package. To begin with, make sure the Java version is no older than 1.4. Otherwise, download a newer version of JAVA from `java.sun.com`.

Install RSmatch1.0 If the input data are RNA structures, follow these instructions to install and run RSmatch1.0.



http://aria.njit.edu/rnacenter/RSmatch/index.html

RSmatch: RNA Secondary Structure Matcher

Jason T. L. Wang (wangj@njit.edu)
& Jianghui Liu (jl7@oak.njit.edu)
Department of Computer Science
New Jersey Institute of Technology
http://web.njit.edu/~wangj

Bin Tian (btian@umdnj.edu)
Department of Biochemistry and Molecular Biology
University of Medicine and Dentistry of New Jersey
http://exon.umdnj.edu

Motivation

Ribonucleic acid (RNA) plays an important role in gene regulatory networks. Many functions of RNA are attributable to the structural conservation in some local regions, called RNA motifs. For many non-coding RNAs, i.e. mRNAs, their global structure conservation is observed. Studying and understanding these structural properties is therefore an important subject in RNA research. The package described here takes into account both sequence and structural information of RNA, allowing alignment of structure-annotated RNA sequences. The package can be applied to RNA motif discovery and database search.

In the current alpha version 1.0, RSmatch provides four functions: (1) regular database search, (2) multiple structure alignment, (3) iterative database search, and (4) pairwise sequence alignment.

1. For a regular database search, the package finds RNA structures in a database that locally or globally match a given (usually small) query structure. The found local regions may become candidate sites containing an RNA motif. This function can also be used to detect motif occurrences in an RNA structure when the query structure is a known motif.
2. The function of multiple structure alignment constructs a multiple local alignment for a given set of RNA structures, by progressively expanding the alignment at each stage. This is a useful tool when a small set of RNAs are functionally

Figure 7.1 The website of RSmatch software package.

1. Download RSmatch1.0 from

   `http://aria.njit.edu/rnacenter/RSmatch/RSmatch.tar.`

2. Extract the tar file to your installation directory, e.g. `/home/RNA`, by typing

   ```
   $tar xvf RSmatch.tar
   ```

3. A directory named `release` under `/home/RNA` will appear. Switch to it by typing

   ```
   $cd release
   ```

4. Type `$RSmatch` to run the program.

Install Vienna RNA v1.4 and RSmatch1.0 If the input data are RNA sequences in the FASTA format, use below instructions to install RSmatch1.0 and Vienna RNA package v1.4.

1. Download Vienna RNA package v1.4 and put it under the `/home/RNA` directory.

2. Unpack the Vienna RNA package by typing

   ```
   $gunzip < ViennaRNA-1.4.tar.gz | tar xvf -
   ```

3. A directory named "ViennaRNA-1.4" under `/home/RNA` will appear. Switch to it by typing

   ```
   $cd ViennaRNA-1.4
   ```

4. Install the Vienna software by typing

   ```
   $make all; make install
   ```

5. Set up the environment variable `VIENNA_HOME`. If the command shell is `bash`, add `export VIENNA_HOME = /home/RNA/ViennaRNA-1.4` to `.bashrc` file. If `csh` is used , add `setenv VIENNA_HOME = /home/RNA/ViennaRNA-1.4` to the `.cshrc` file. It is needed to log out and log in again to make it effective.

6. Install and run RSmatch1.0 by following the instructions above. RSmatch1.0 will automatically invoke Vienna RNA v1.4 to fold the input sequences into structures and then align the structures.

7.1.3 Input and Output

There are two types of input data. The first type is the nested parenthesized notation representing an RNA secondary structure. For each structure, it has three lines: header line, primary sequence line and structure notation line. A sample structure is like this:

```
>NM_003234  Homo sapiens transferrin receptor (p90, CD71) (TFRC), mRNA
GCTTTCTGTCCTTTTGGCACTGAGATATTTTTTTATCAGTGACAGAGTTCACTATAAATGTATCGGAAGC
((((((.((((....)).))...((((.....((((.(((((......)))))))))...))))))))))
```

The second type is the FASTA format for RNA sequences. For the sequence data, RSmatch1.0 will automatically invoke Vienna RNA v1.4 to fold the sequences into structures and then align the structures. A sample sequence in the FASTA format is like this:

```
>NM_123456 Homo sapiens transferrin receptor (p90, CD71) (TFRC)
GCTTTCTGTCCTTTTGGCACTGAGATATTTATTGTTTATTTATCAGTGACAGAGTTCACTATAAATGGTG
TTTTTTTAATAGAATATAATTATCGGAAGC
```

The output of RSmatch1.0 gives detailed alignment information. The Stockholm format is adopted to display the output of multiple structure alignment.

7.1.4 Options

You can find the general syntax of the command by typing `RSmatch`. The general syntax in fact is:

```
RSmatch [options] General options:
  -p [dsearch | isearch | mrsa | prsa]
    choose a program:
      dsearch     simple database search;
      isearch     iterative database search;
      mrsa        multiple RNA structure alignment;
      prsa        pair-wise RNA structure alignment;
  -D <database>   FASTA-formatted sequence database.
  -d <database>   secondary structure database.
  -g <penalty>    gap penalty.
  -o <output>     output file; default is 'result.out'.
  -r <range>      range of folding free energy (kcal/mol) used to
                  select alternative RNA structures; default is 0.
  -S <ratio>      sliding step length, expressed as a ratio of
                  <W_length>; default is 0.5.
  -W <W_length>   sliding window size; default is 100 nt.
Options for 'dsearch' and 'isearch':
  -n <topN>       output top 'topN' hits.
  -Q <query>      query sequence in FASTA format.
  -q <query>      query structure.
Options for 'dsearch' and 'prsa':
  -s <score_matrix>   file containing position independent score
                      matrices; default is 'scoreMat.structure'.
Options for 'dsearch':
  -G <global alignment>
     T:     global alignment
```

```
      F:     local alignment
      default: F
  -m <query type>      query type:
                         0: real structure without IUB code;
                         1: pattern structure containing IUB code.
                         default: 0
Options for 'isearch':
  -R <repeat>          number of iterations.
Options for 'prsa':
  -F <factor>          the window-size decreasing rate. A series of
                       window sizes are generated for folding sequences.
                       The <factor> is the ratio of two contiguous
                       window sizes.
```

Here are some more examples with commentary:

- `RSmatch -p dsearch -d test.structDB -q query.struct`

  will use the secondary structure in query.struct to search against the structure database `test.structDB`. The output is in the default file `result.out`.

- `RSmatch -p dsearch -d test.structDB -q motif.ire1 -m 1`

  will use the pattern structure in `motif.ire1` to search against the structure database `test.structDB` and store the result in the default file `result.out`.

- `RSmatch -p mrsa -d test.structDB`

  will construct multiple structure alignment for the structures in `test.structDB`. The output is stored in the default file `result.out`.

- `RSmatch -p isearch -d test.structDB -q query.struct -R 2`

  will perform an iterative database search with two iterations.

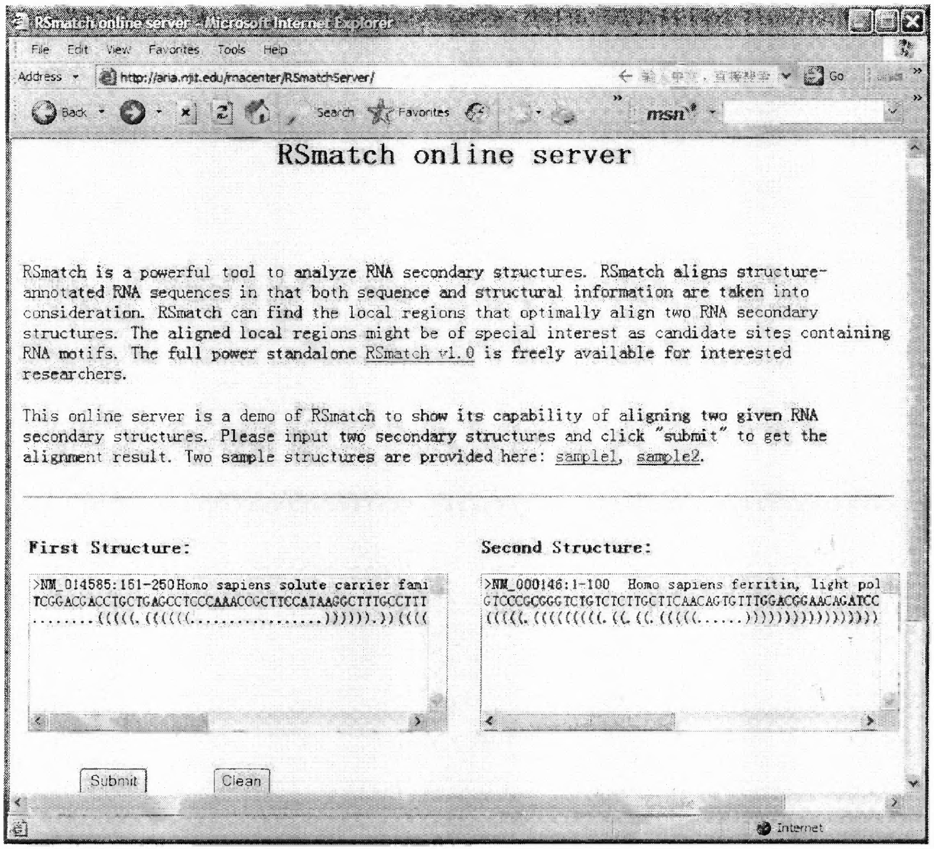
- `RSmatch -p prsa -D nanos.seq -W 120 -S 0.2 -F 0.5`

  will compare the first two sequences in `nanos.seq`. The initial folding window size is $120nt$ and the sliding step length is 0.2 times the window size. Subsequent window sizes decrease at the ratio of 0.5.

- `RSmatch -p dsearch -D test.seqDB -Q query.seq -r 1.7`

  will perform a database search using both a sequence query and a sequence database. The query structure will be obtained by folding the sequence in `query.seq`; the structure database will be constructed by folding the sequences in `test.seqDB`. Folded structures whose energy are not 1.7kcal/mol higher than that of the optimal energy structure are kept.

7.2 RSmatch Online Server


RSmatch online server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Back
Search
Favorites
msn
Go
Internet

Address: http://aria.njit.edu/macenter/RSmatchServer/

RSmatch online server

RSmatch is a powerful tool to analyze RNA secondary structures. RSmatch aligns structure-annotated RNA sequences in that both sequence and structural information are taken into consideration. RSmatch can find the local regions that optimally align two RNA secondary structures. The aligned local regions might be of special interest as candidate sites containing RNA motifs. The full power standalone RSmatch v1.0 is freely available for interested researchers.

This online server is a demo of RSmatch to show its capability of aligning two given RNA secondary structures. Please input two secondary structures and click "submit" to get the alignment result. Two sample structures are provided here: sample1, sample2.

First Structure:

```
>NM_014585:151-250Homo sapiens solute carrier fami
TCGGACGACCTGCTGAGCCTCCCAAACCGCTTCCATAAGGCTTTGCCTTT
........(((((.((((((.................)))))).))((((
```

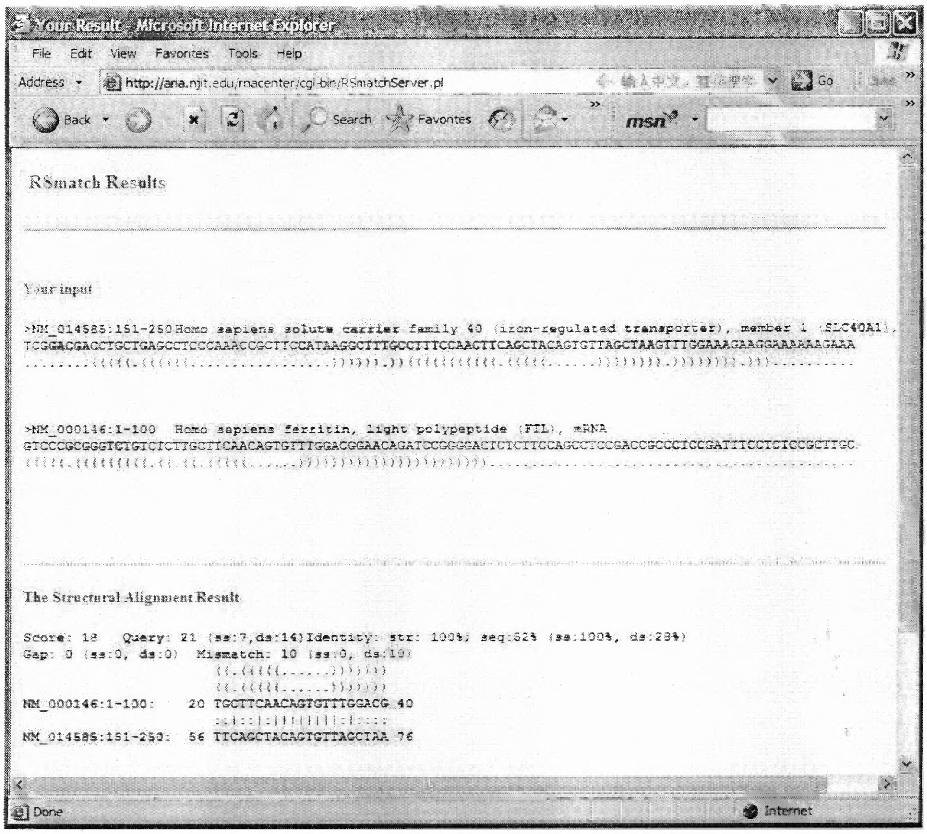
Second Structure:

```
>NM_000146:1-100  Homo sapiens ferritin, light pol
GTCCCGCGGGTCTGTCTCTTGCTTCAACAGTGTTTGGACGGAACAGATCC
(((((.(((((((((.((.((.(((((......)))))))))))))))))
```

Submit   Clean

Figure 7.2 The input screenshot of RSmatch online server.

A thin online RSmatch server has been setup to demonstrate the power of RSmatch in aligning two given RNA secondary structures. User can enter or paste secondary structures and run the online server to get the result instantly. Some screenshots are provided to show its usage (ref Fig.7.2, Fig.7.3).


Your Result - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://ana.njit.edu/rnacenter/cgi-bin/RSmatchServer.pl
Go
Back
Search
Favorites
msn
Done
Internet

RSmatch Results

Your input

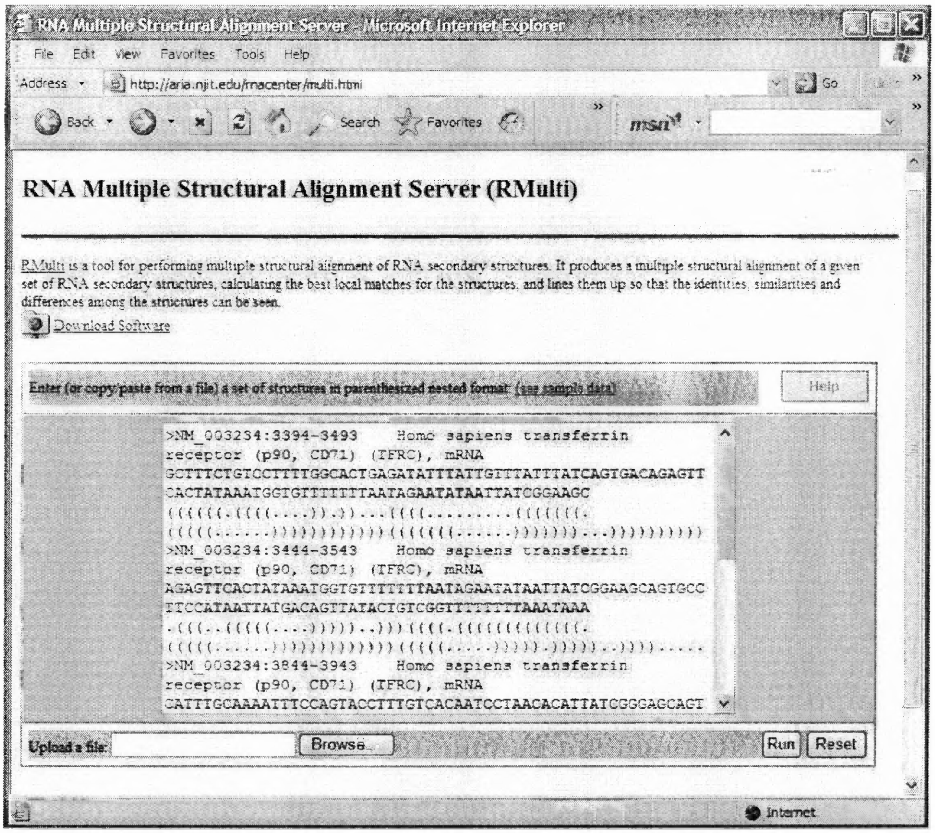
```
>NM_014585:151-250 Homo sapiens solute carrier family 40 (iron-regulated transporter), member 1 (SLC40A1),
TCGGACGACCTGCTGAGCCTCCCAAACCGCTTCCATAAGGCTTTGCCTTTCCAACTTCAGCTACAGTGTTAGCTAAGTTTGGAAAGAAGGAAAAAAGAAA
........((((.((((((....................)))))).))((((((((((.(((((......))))))))).)))))))).))..........

>NM_000146:1-100  Homo sapiens ferritin, light polypeptide (FTL), mRNA
GTCCCGCGGGTCTGTCTCTTGCTTCAACAGTGTTTGGACGGAACAGATCCGGGGACTCTCTTCCAGCCTCCGACCGCCCTCCGATTTCCTCTCCGCTTGC
(((((.(((((((((.((.((.(((((......)))))))))))))))))))))))............................................
```

The Structural Alignment Result

```
Score: 18   Query: 21 (ss:7,ds:14)Identity: str: 100%; seq:52% (ss:100%, ds:28%)
Gap: 0 (ss:0, ds:0)  Mismatch: 10 (ss:0, ds:10)
                          ((.(((((......)))))))
                          ((.(((((......)))))))
NM_000146:1-100:      20 TGCTTCAACAGTGTTTGGACG 40
                          ::|::|:||||||||:|::::
NM_014585:151-250:   56 TTCAGCTACAGTGTTAGCTAA 76
```

Figure 7.3 The result screenshot of RSmatch online server.


RNA Multiple Structural Alignment Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://aria.njit.edu/rnacenter/multi.html
Go
Back
Search
Favorites
msn
Help
Browse...
Run
Reset
Internet

RNA Multiple Structural Alignment Server (RMulti)

RMulti is a tool for performing multiple structural alignment of RNA secondary structures. It produces a multiple structural alignment of a given set of RNA secondary structures, calculating the best local matches for the structures, and lines them up so that the identities, similarities and differences among the structures can be seen.

Download Software

Enter (or copy/paste from a file) a set of structures in parenthesized nested format: (see sample data)

```
>NM_003234:3394-3493    Homo sapiens transferrin
receptor (p90, CD71) (TFRC), mRNA
GCTTTCTGTCCTTTTGGCACTGAGATATTTATTGTTTATTTATCAGTGACAGAGTT
CACTATAAATGGTGTTTTTTTTAATAGAATATAATTATCGGAAGC
((((((.((((....)).))...((((.........(((((((.
((((((.....))))))))))))((((((((......)))))))...))))))))))
>NM_003234:3444-3543    Homo sapiens transferrin
receptor (p90, CD71) (TFRC), mRNA
AGAGTTCACTATAAATGGTGTTTTTTTTAATAGAATATAATTATCGGAAGCAGTGCC
TTCCATAATTATGACAGTTATACTGTCGGTTTTTTTTTAAATAAA
.(((..((((((....))))))..)))((((.(((((((((((((.
((((((.....)))))))))))))((((((.....)))))).))))))..)))).....
>NM_003234:3844-3943    Homo sapiens transferrin
receptor (p90, CD71) (TFRC), mRNA
CATTTGCAAAATTTCCAGTACCTTTGTCACAATCCTAACACATTATCGGGAGCAGT
```

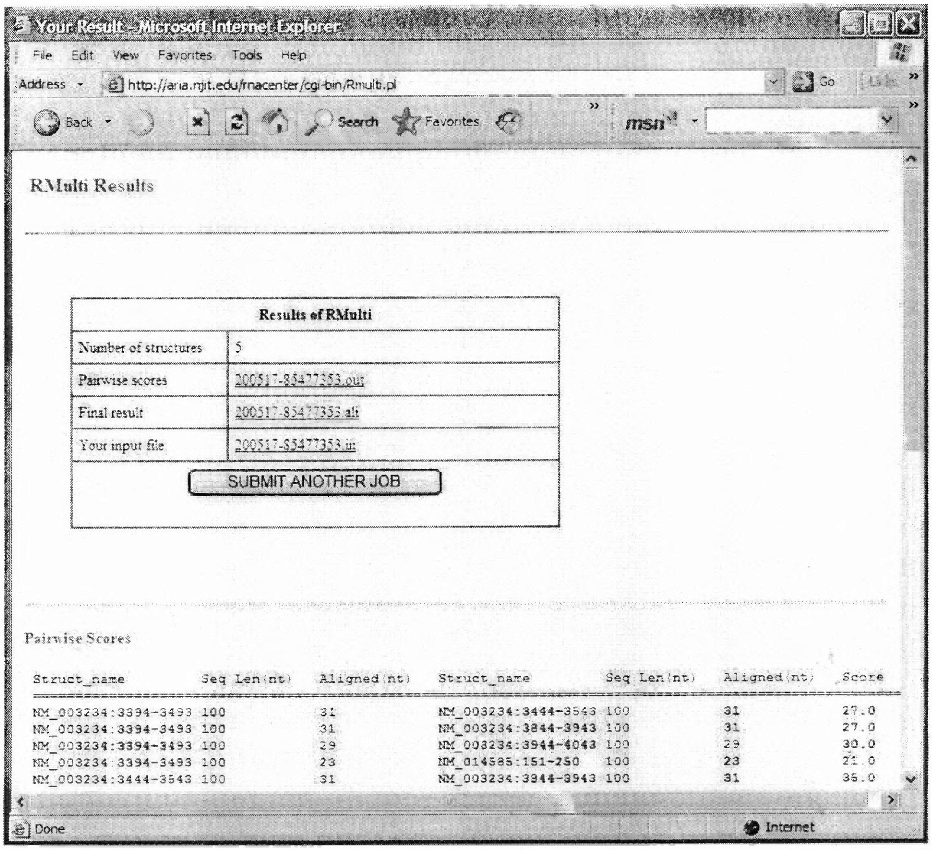
Upload a file:

Figure 7.4 The multiple structural alignment server.

7.3 Rmult Multiple Structural Alignment Server

There is another important online server - Rmulti multiple structure alignment server. Using Rmulti (ref Fig.7.4), functionally or structurally related structures can be aligned with each other at interesting local motif region. This is a valuable tool for locating motif site among a set of structures.

Specifically, RMulti is a tool for performing multiple structural alignment of RNA secondary structures. It produces a multiple structural alignment of a given small set of RNA secondary structures, calculating the best local matches for the structures, and lines them up so that the identities, similarities and differences among the structures can be seen. Meaningful screenshots can be found in Fig.7.5 and Fig.7.6.


Your Result - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
http://aria.njit.edu/rnacenter/cgi-bin/Rmulti.pl
Go
Back
Search
Favorites
msn
Done
Internet

RMulti Results

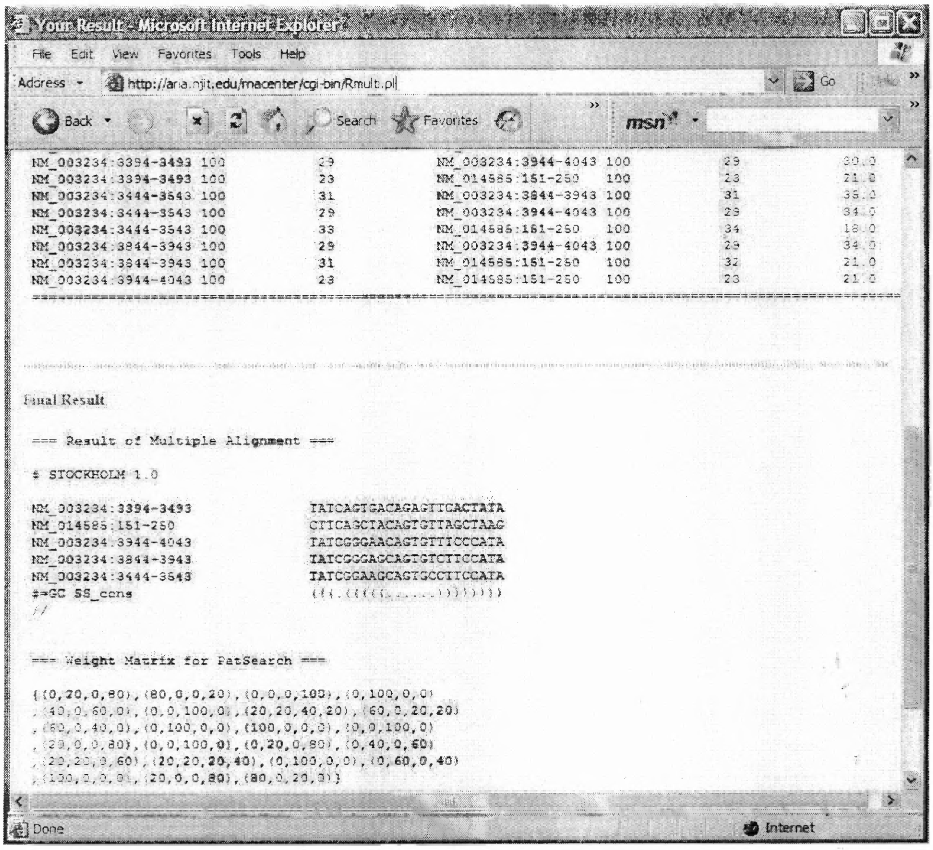
Results of RMulti	
Number of structures	5
Pairwise scores	200517-85477353.out
Final result	200517-85477353.aln
Your input file	200517-85477353.in

SUBMIT ANOTHER JOB

Pairwise Scores

```
Struct_name              Seq Len(nt)   Aligned(nt)   Struct_name              Seq Len(nt)   Aligned(nt)   Score
==============================================================================================================
NM_003234:3394-3493 100              31             NM_003234:3444-3543 100              31             27.0
NM_003234:3394-3493 100              31             NM_003234:3844-3943 100              31             27.0
NM_003234:3394-3493 100              29             NM_003234:3944-4043 100              29             30.0
NM_003234:3394-3493 100              23             NM_014585:151-250   100              23             21.0
NM_003234:3444-3543 100              31             NM_003234:3844-3943 100              31             35.0
```

Figure 7.5 The first half of output from multiple structural alignment server.


Your Result - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
http://aria.njit.edu/rnacenter/cgi-bin/Rmulti.pl
Go
Back
Search
Favorites
msn
Done
Internet

```
NM_003234:3394-3493 100          29         NM_003234:3944-4043 100          29          30.0
NM_003234:3394-3493 100          23         NM_014585:151-250   100          23          21.0
NM_003234:3444-3543 100          31         NM_003234:3844-3943 100          31          35.0
NM_003234:3444-3543 100          29         NM_003234:3944-4043 100          23          34.0
NM_003234:3444-3543 100          33         NM_014585:151-250   100          34          18.0
NM_003234:3844-3943 100          29         NM_003234:3944-4043 100          29          34.0
NM_003234:3844-3943 100          31         NM_014585:151-250   100          32          21.0
NM_003234:3944-4043 100          23         NM_014585:151-250   100          23          21.0
==============================================================================================
```

Final Result

```
=== Result of Multiple Alignment ===

# STOCKHOLM 1.0

NM_003234:3394-3493            TATCAGTGACAGAGTTCACTATA
NM_014585:151-250              CTTCAGCTACAGTGTTAGCTAAG
NM_003234:3944-4043            TATCGGGAACAGTGTTTCCCATA
NM_003234:3844-3943            TATCGGGAGCAGTGTCTTCCATA
NM_003234:3444-3543            TATCGGAAGCAGTGCCTTCCATA
#=GC SS_cons                   (((.(((((......))))))))
//


=== Weight Matrix for PatSearch ===

{(0,20,0,80),(80,0,0,20),(0,0,0,100),(0,100,0,0)
,(40,0,60,0),(0,0,100,0),(20,20,40,20),(60,0,20,20)
,(60,0,40,0),(0,100,0,0),(100,0,0,0),(0,0,100,0)
,(20,0,0,80),(0,0,100,0),(0,20,0,80),(0,40,0,60)
,(20,20,0,60),(20,20,20,40),(0,100,0,0),(0,60,0,40)
,(100,0,0,0),(20,0,0,80),(80,0,20,0)}
```

Figure 7.6 The second half of output from multiple structural alignment server.

CHAPTER 8

CONCLUSION AND FUTURE WORKS

The most important and interesting application is to discover novel motifs. As presented in the experiments, the proposed technique is capable of discovering motifs in an efficient way, at least in the case of finding IRE motif. However, one drawback in the proposed framework is that the structure prediction tools (such as MFOLD [78], RNA Vienna package [39] may not give rise to correct RNA structures. A solution of this is to choose suboptimal structures in addition to the optimal one to increase the chance of obtaining correct structures.

In this doctoral thesis, efficient algorithms for aligning RNA secondary structures and mining unknown RNA motifs are presented. As the major contribution, a structure alignment algorithm, which combines both primary and secondary structure information, can find the optimal alignment between two given structures where one of them could be either a pattern structure of a known motif or a real query structure and the other be a subject structure.

Motivated by widely used algorithms for RNA folding, the proposed algorithm decomposes an RNA secondary structure into a set of atomic structural components that can be further organized in a tree model to capture the structural particularities. The novel structure alignment algorithm is implemented using dynamic programming techniques coupled by position-independent scoring matrices. The algorithm can find the optimal global and local alignment between two RNA secondary structures at quadratic time complexity. When applied to searching a structure database, the algorithm can find similar RNA substructures and therefore can be used to identify functional RNA motifs. Extension of the algorithm has also been accomplished to deal with position-dependent scoring matrix in the purpose of aligning multiple structures.

All algorithms have been implemented in a package under the name *RSmatch* and applied to searching mRNA UTR structure database and mining RNA motifs. The experimental results showed high efficiency and effectiveness of the proposed techniques.

There exist several promising future works in RNA structure field and other related applications. One is the statistical significance assessment of high scoring structures. This is a very complicated task compared with what had been done in p-value calculation for protein sequence analysis [79]. It might be possible to do some approximation of the real distribution. But the theoretical analysis remains in mystery.

Motivated by the statistical methods of assessing results in sequence alignment [79], efforts has been made to develop scores of the structure database search with known probabilistic distributions. The score distribution seemed close to be normal. However since the scoring scheme is still at its preliminary stage and much need to be learned about the RNA structure database presented in this study, more elaborate statistical assessment of the search results will be developed in the future.

BIBLIOGRAPHY

[1] T. Smith and M. Waterman, "The identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.

[2] B. Shapiro and K. Zhang, "Comparing multiple RNA secondary structures using tree comparisons," *Computer Application in Bioscience*, vol. 6, pp. 309–318, 1990.

[3] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. New York, NY: Combridge Press, 1997.

[4] D. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Communications of the ACM*, vol. 18, pp. 341–343, 1975.

[5] E. Mayers and W. Miller, "Optimal alignments in linear space," *Computer Application in Bioscience*, vol. 4, pp. 11–17, 1988.

[6] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "A basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, pp. 403–410, 1990.

[7] S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, pp. 3389–3402, 1997.

[8] W. Pearson and D. Lipman, "Improved tools for biological sequence comparison," *Proceedings of the National Academy of Science USA*, pp. 2444–2448, 1988.

[9] J. Thompson, D. Higgins, and T. Gibson, "CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties, and weight matrix choice," *Nucleic Acids Research*, vol. 22, pp. 4673–4680, 1994.

[10] D. Gautheret and A. Lambert, "Direct RNA motif definition and identication from multiple sequence alignments using secondary structure profiles," *Journal of Molecular Biology*, vol. 313, pp. 1003–1011, 2001.

[11] P. Bork, C. Ouzounis, C. Sander, M. Scharf, R. Schneider, and E. Sonnhammer, "Comprehensive sequence analysis of the 182 predicted open reading frames of yeast chromosome III," *Protein Science*, vol. 1, pp. 1677–1690, 1992.

[12] P. Green, D. Lipman, L. Hillier, R. Waterson, D. States, and J.-M. Claverie, "Ancient conserved regions in new gene sequences and the protein databases," *Science*, vol. 259, pp. 1711–1716, 1993.

[13] S. Needleman and C. Wunsch, "A general method applicable to the search for siimilarities in the amino-acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[14] R. Wagner and M. Fischer, "The string-to-string correction problem," *Journal of the ACM*, vol. 21, no. 1, pp. 168–178, 1974.

[15] S. Eddy, "Computational genomics of noncoding RNA genes," *Cell*, vol. 109, pp. 137–140, 2002.

[16] S. Kuersten and E. Goodwin, "The power of 3'UTR: translational control and development," *Nature Reviews Genetics*, vol. 4, no. 8, pp. 626–637, 2003.

[17] G. Wilkie, K. Dickson, and N. Gray, "Regulation of mRNA translation by 5'- and 3'-UTR-binding factors," *Trends in Biochemical Sciences*, vol. 28, no. 4, pp. 182–188, 2003.

[18] F. Mignone, C. Gissi, S. Liuni, and G. Pesole, "Untranslated regions of mRNAs," *Genome Biology*, vol. 3, no. 3, pp. 0004.0001–0004.0010, 2002.

[19] P. Bengert and T. Dandekar, "A software tool-box for analysis of regulatory RNA elements," *Nucleic Acids Research*, vol. 31, pp. 3441–3445, 2003.

[20] J. Gorodkin, L. Heyer, and G. Stormo, "Finding the most significant common sequence and structure motifs in a set of RNA sequences," *Nucleic Acids Research*, vol. 25, pp. 3724–3732, 1997.

[21] G. Grillo, F. Licciulli, S. Liuni, E. Sbisa, and G. Pesole, "PatSearch: a program for the detection of patterns and structural motifs in nucleotide sequences," *Nucleic Acids Research*, vol. 31, pp. 3608–3612, 2003.

[22] T. Macke, D. Ecker, R. Gutell, D. Gautheret, D. Case, and R. Sampath, "RNAMotif, an RNA secondary structure definition and search algorithm," *Nucleic Acids Research*, vol. 29, pp. 4724–4735, 2001.

[23] G. Pavesi, G. Mauri, and G. Pesole, "An algorithm for finding conserved secondary structure motifs in unaligned RNA sequences," *Journal of Computer Science and Technology*, vol. 19, pp. 2–12, 2004.

[24] G. Pavesi, G. andMauri, M. Stefani, and G. Pesole, "RNAProfile: an algorithm for finding conserved secondary structure motifs in unaligned RNA sequences," *Nucleic Acids Research*, vol. 32, pp. 3258–3269, 2004.

[25] G. Pesole, S. Liuni, and M. D'Souza, "PatSearch: a pattern matcher software that finds functional elements in nucleotide and protein sequences and assess their statistical significance," *Bioinformatics*, vol. 16, no. 5, pp. 439–450, 2000.

[26] A. Krogh, M. Brown, I. Mian, K. Sjölander, and D. Haussler, "Hidden markov models in computational biology: Application to protein modeling," *Journal of Molecular Biology*, vol. 235, pp. 1501–1531, 1994.

[27] J. Jaeger, D. Turner, and M. Zuker, "Improved predictions of secondary structures for RNA," *Proceedings of National Academy of Science USA*, vol. 86, no. 20, pp. 7706–7710, 1989.

[28] M. Zuker, "On finding all suboptimal foldings of an RNA molecule," *Science*, vol. 244, pp. 48–52, 1989.

[29] M. Zuker, "RNA folding prediction: the continued need for interaction between biologists and mathematicians," *Lectures on Mathematics in the Life Sciences*, vol. 17, pp. 86–123, 1986.

[30] T. Jiang, G. Lin, B. Ma, and K. Zhang, "A general edit distance between RNA structures," *Journal of Computational Biology*, vol. 9, no. 2, pp. 371–388, 2002.

[31] I. Hofacker, W. Fontana, P. Stadler, L. Bonhoeffer, M. Tacker, and P. Schuster, "Fast folding and comparison of RNA secondary structures," *Monatshefte für Chemie*, vol. 125, pp. 167–188, 1994.

[32] B. Shapiro, "An algorithm for comparing multiple RNA secondary structures," *Computer Application in Bioscience*, vol. 4, pp. 387–393, 1988.

[33] W. Schmitt and M. Waterman, "Linear trees and RNA secondary structure," *Discrete Applied Mathematics*, vol. 51, pp. 317–323, 1994.

[34] G. Collins, S. Le, and K. Zhang, "A new algorithm for computing similarity between RNA structures," *Information Sciences*, vol. 139, pp. 59–77, 2001.

[35] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khana, and S. Eddy, "Rfam: an RNA family database," *Nucleic Acids Research*, vol. 31, pp. 439–441, 2003.

[36] V. Ambros, B. Bartel, D. Bartel, C. Burge, J. Carrington, X. Chen, G. Dreyfuss, S. Eddy, S. Griffiths-Jones, and M. Marshall, "A uniform system for microRNA annotation," *RNA*, vol. 9, pp. 277–279, 2003.

[37] G. Pesole, S. Liuni, G. Grillo, F. Licciulli, F. Mignone, C. Gissi, and C. Saccone, "UTRdb and UTRsite: specialized databases of sequences and functional elements of 5' and 3' untranslated regions of eukaryotic mRNAs," *Nucleic Acids Research*, vol. 30, pp. 335–340, 2002.

[38] M. W.F. and D. R.J., "Histone mRNA expression: multiple levels of cell cycle regulation and important developmental consequences," *Current Opinion on Cell Biology*, vol. 14, pp. 692–699, 2002.

[39] I. Hofacker, "Vienna RNA secondary structure server," *Nucleic Acids Research*, vol. 31, pp. 3429–3431, 2003.

[40] P. Schuster, W. Fontana, P. Stadler, and I. Hofacker, "From sequences to shapes and back: a case study in RNA secondary structures," in *Proceedings of the Royal Society London B: Biological Science*, vol. 255, pp. 279–284, 1994.

[41] E. Rivas and S. Eddy, "A dynamic programming algorithm for rna structure prediction including pseudoknots," *Journal of Molecular Biology*, vol. 285, no. 5, pp. 2053–2068, 1999.

[42] B. Gulko and D. Haussler, "Using multiple alignments and phylogenetic trees to detect RNA secondary structure," in *Pacific Symposium of Biocomputing*, pp. 350–367, 1996.

[43] V. Akmaev, S. Kelley, and G. Stormo, "A phylogenetic approach to RNA structure prediction," in *Proceedings of the International Conference on Intelligent Systems in Molecular Biology*, pp. 10–17, AAAI/MIT Press, 1999.

[44] K. Bjarne and H. Jotun, "Pfold: RNA secondary structure prediction using stochastic context-free grammars," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3423–3428, 2003.

[45] I. Hofacker, M. Fekete, and P. Stadler, "Secondary structure prediction for aligned RNA sequences," *Journal of Molecular Biology*, vol. 319, no. 5, pp. 1059–1066, 2002.

[46] D. Sankoff, "Simultaneous solution of the RNA folding, alignment and protosequence problems," *SIAM Journal of Applied Mathematics*, vol. 45, pp. 810–825, 1985.

[47] J. Gorodkin, S. Stricklin, and G. Stormo, "Discovering common stem-loop motifs in unaligned RNA sequences," *Nucleic Acids Research*, vol. 29, no. 10, pp. 2135–2144, 2001.

[48] D. Mathews and D. Turner, "Dyalign: an algorithm for finding the secondary structure common to two RNA sequences," *Journal of Molecular Biology*, vol. 317, pp. 191–203, 2002.

[49] O. Perriquet, H. Touzet, and M. Dauchet, "Finding the common structure shared by two homologous RNAs," *Bioinformatics*, vol. 19, no. 1, pp. 108–118, 2003.

[50] Y. Ji, X. Xu, and G. Stormo, "A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences," *Bioinformatics*, vol. 20, no. 10, pp. 1591–1602, 2004.

[51] C. Notredame, E. OBrien, and D. Higgins, "RAGA: RNA sequence alignment by genetic algorithm," *Nucleic Acids Research*, vol. 25, pp. 4570–4580, 1997.

[52] J. Kim, J. Cole, and S. Pramanik, "Alignment of possible secondary structures in multiple RNA sequences using simulated annealing," *Comput. Appl. Biosci*, vol. 12, pp. 259-267, 1996.

[53] J. Chen, S. Le, and J. Maizel, "Prediction of common secondary structures of RNAs: a genetic algorithm approach," *Nucleic Acids Research*, vol. 28, pp. 991–999, 2000.

[54] G.-H. Lin, M. Bin, and Z. Kaizhong, "Edit distance between two RNA structures," in *Proceedings of RECOMB 2001*, (Montreal, Canada), pp. 211–220, 2001.

[55] M. Hochsmann, T. Toller, R. Giegerich, and S. Kurtz, "Local similarity in RNA secondary structures," in *Proceedings of the IEEE Bioinformatics conference 203*, pp. 159–168, IEEE, 2003.

[56] Y. Sakakibara, M. Brown, R. Hughey, I. Mian, K. Sjölander, R. Underwood, and D. Haussler, "Stochastic context-free grammars for tRNA modeling," *Nucleic Acids Research*, vol. 22, pp. 5112–5120, 1994.

[57] S. Eddy and R. Durbin, "RNA sequence analysis using covariance models," *Nucleic Acids Research*, vol. 22, pp. 2079–2088, 1994.

[58] L. T. and E. S.R., "A computational screen for methylation guide snoRNAs in yeast," *Science*, vol. 283, pp. 1168–1171, 1999.

[59] K. R.J. and E. S.R., "RSEARCH: finding homologs of single structured RNA sequences," *BMC Bioinformatics*, vol. 4, no. 44, 2003.

[60] H. I. and R. G.M., "Pairwise RNA structure comparison with stochastic context-free grammars," in *Pacific Symposium of Biocomputing*, pp. 163–174, 2002.

[61] A. Laferriere, D. Gautheret, and R. Cedergren, "An RNA pattern matching program with enhanced performance and portability," *Comput. Appl. Biosci.*, vol. 10, pp. 211–212, 1994.

[62] T. Bakheet, M. Frevel, B. Williams, W. Greer, and K. Khabar, "ARED: human AU-rich element-containing mRNA database reveals an unexpectedly diverse functional repertoire of encoded proteins," *Nucleic Acids Research*, vol. 29, pp. 246–254, 2001.

[63] C. Chen and A. Shyu, "AU-rich elements: characterization and importance in mRNA degradation," *Trands in Biochemistry Science*, vol. 20, pp. 465–470, 1995.

[64] C. Wilusz and J. Wilusz, "Bringing the role of mRNA decay in the control of gene expression into focus," *Trands in Genetics*, vol. 20, pp. 491–497, 2004.

[65] D. Bartel, "MicroRNAs: genomics, biogenetics, mechanism, and function," *Cell*, vol. 116, pp. 281–291, 2004.

[66] M. Blanchette and M. Tompa, "Discovery of regulatory elements by a computational method for phylogenetic footprinting," *Genome Research*, vol. 12, pp. 739–748, 2002.

[67] D. Boffelli, J. McAuliffe, D. Ovcharenko, K. Lewis, I. Ovcharenko, L. Pachter, and E. Rubin, "Phylogenetic shadowing of primate sequences to find functional regions of the human genome," *Science*, vol. 299, pp. 1391–1394, 2003.

[68] W. Fairbrother, R. Yeh, P. Sharp, and C. Burge, "Predictive identification of exonic splicing enhances in human genes," *Science*, vol. 297, pp. 1007–1013, 2002.

[69] L. Marino-Ramirez, J. Spouge, G. Kanga, and D. Landsman, "Statistical analysis of over-represented words in human promoter sequences," *Nucleic Acids Research*, vol. 32, pp. 949–958, 2004.

[70] A. Smith, P. Sumazin, and M. Zhang, "Identifying tissue-selective transcription factor binding sites in vertebrate promoters," *Proceedings of National Academy of Science USA*, vol. 102, pp. 1560–1565, 2005.

[71] X. Xie, J. Lu, E. Kulbokas, T. Golub, V. Mootha, K. Lindblad-Toh, E. Lander, and M. Kellis, "Systematic discovery of regulatory motifs in human promoters and 3'UTRs by comparison of several mammals," *Nature*, vol. 434, pp. 338–345, 2005.

[72] V. Akmaev, S. Kelley, and G. Stormo, "Phylogenetically enhanced statistical tools for RNA structure prediction," *Bioinformatics*, vol. 16, pp. 501–512, 2000.

[73] E. Rivas, R. Klein, T. Jones, and S. Eddy, "Computational identification of noncoding RNAs in e.coli by comparative genomics," *Current Biology*, vol. 11, pp. 1369–1373, 2001.

[74] S. Washietl and I. Hofacker, "Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics," *Journal of Molecular Biology*, vol. 342, pp. 19–30, 2004.

[75] J. Liu, J. Wang, J. Hu, and B. Tian, "A method for aligning RNA secondary structures and its application to RNA motif detection," *BMC Bioinformatics*, vol. 6, no. 89, 2005.

[76] K. Pruitt, K. Katz, H. Sicotte, and D. Maglott, "Introducing refseq and locuslink: curated human genome resources at the NCBI," *Trends in Genetics*, vol. 16, pp. 44–47, 2000.

[77] B. Tian, J. Hu, H. Zhang, and C. Lutz, "A large-scale analysis of mRNA polyadenylation of human and mouse genes," *Nucleic Acids Research*, vol. 33, pp. 201–212, 2005.

[78] D. Mathews, J. Sabina, M. Zuker, and D. Turner, "Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure," *Journal of Molecular Biology*, vol. 288, pp. 911–940, 1999.

[79] S. Karlin and S. Altschul, "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes," *Proc. Natl. Acad. Sci. USA*, vol. 87, pp. 2264–2268, 1990.