ABSTRACT

DESIGN AND IMPLEMENTATION OF A CYBERINFRASTRUCTURE FOR RNA MOTIF SEARCH, PREDICTION AND ANALYSIS

by
Dongrong Wen

RNA secondary and tertiary structure motifs play important roles in cells. However, very few web servers are available for RNA motif search and prediction. In this dissertation, a cyberinfrastructure, named RNAcyber, capable of performing RNA motif search and prediction, is proposed, designed and implemented.

The first component of RNAcyber is a web-based search engine, named RmotifDB. This web-based tool integrates an RNA secondary structure comparison algorithm with the secondary structure motifs stored in the Rfam database. With a user-friendly interface, RmotifDB provides the ability to search for ncRNA structure motifs in both structural and sequential ways. The second component of RNAcyber is an enhanced version of RmotifDB. This enhanced version combines data from multiple sources, incorporates a variety of well-established structure-based search methods, and is integrated with the Gene Ontology. To display RmotifDB's search results, a software tool, called RSview, is developed. RSview is able to display the search results in a graphical manner.

Finally, RNAcyber contains a web-based tool called Junction-Explorer, which employs a data mining method for predicting tertiary motifs in RNA junctions. Specifically, the tool is trained on solved RNA tertiary structures obtained from the Protein Data Bank, and is able to predict the configuration of coaxial helical stacks and families (topologies) in RNA junctions at the secondary structure level. Junction-

Explorer employs several algorithms for motif prediction, including a random forest classification algorithm, a pseudoknot removal algorithm, and a feature ranking algorithm based on the gini impurity measure. A series of experiments including 10-fold cross-validation has been conducted to evaluate the performance of the Junction-Explorer tool. Experimental results demonstrate the effectiveness of the proposed algorithms and the superiority of the tool over existing methods. The RNAcyber infrastructure is fully operational, with all of its components accessible on the Internet.

DESIGN AND IMPLEMENTATION OF A CYBERINFRASTRUCTURE FOR RNA MOTIF SEARCH, PREDICTION AND ANALYSIS

by
Dongrong Wen

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

January 2012

APPROVAL PAGE

DESIGN AND IMPLEMENTATION OF A CYBERINFRASTRUCTURE FOR RNA MOTIF SEARCH, PREDICTION AND ANALYSIS

Dongrong Wen

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

Dr. James A. M. 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. Christian E. Laing, Committee Member Date
Assistant Professor of Biology, Mathematics and Computer Science, Wilkes University

Dr. Tamar Schlick, Committee Member Date
Professor of Chemistry, Mathematics, and Computer Science, New York University

BIOGRAPHICAL SKETCH

Author: Dongrong Wen

Degree: Doctor of Philosophy

Date: January 2012

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2012

- Master of Science in Computer Science,
  New York University, New York, NY, 2003

- Bachelor of Engineering in Information Engineering and Computer Science,
  Feng Chia University, Taichung, Taiwan, 1998

Major: Computer Science

Presentations and Publications:

Laing,C., Wen,D., Wang,J.T.L. and Schlick,T. (2011) Predicting coaxial helical stacking in RNA junctions. *Nucleic Acids Research*, Accepted.

Wen,D. and Wang,J.T.L. (2010) Structural Search in RNA Motif Databases. *Computational Intelligence and Pattern Analysis in Biology Informatics*, John Wiley & Sons, Inc., Hoboken, New Jersey.

Wen,D. and Wang,J.T.L. (2009) Design of an RNA Structural Motif Database. *International Journal of Computational Intelligence in Bioinformatics and Systems Biology*, **1**, 32-41.

Khaladkar,M., Liu,J., Wen,D., Wang,J.T.L. and Tian,B. (2008) Mining Small RNA Structure Elements in Untranslated Regions of Human and Mouse mRNAs Using Structure-Based Alignment. *BMC Genomics*, **9**:189.

Wen,D. (2008) Design and Implementation of an RNA Secondary Structure Motif Database. *Annual CS Ph.D. Student Research Day*, New Jersey Institute of Technology, Newark, New Jersey.

Wang,J.T.L., Wen,D., Shapiro,B.A., Herbert,K.G., Li,J. and Ghosh,K. (2007) Toward an Integrated RNA Motif Database. *Proceedings of the 4th International Workshop on Data Integration in the Life Sciences (DILS 2007)*, Philadelphia, Pennsylvania, 27-36.

Wang,J.T.L., Wen,D. and Liu,J. (2007) On Comparing and Visualizing RNA Secondary Structures. *Knowledge Discovery in Bioinformatics: Techniques, Methods, and Applications*, John Wiley & Sons, Inc., Hoboken, New Jersey.

Wen,D. (2006) RSview: a RNA comparison and visualization tool. *Annual CS Ph.D. Student Research Day*, New Jersey Institute of Technology, Newark, New Jersey.

Wen,D. (2005) RSmatchView: Comparing and Visualizing RNA Secondary Structures. *1st Annual Research Showcase, Society of Biology*, New Jersey Institute of Technology, Newark, New Jersey.

To my beloved grandma, parents, sister and especially my wife,
for their endless love, support and encouragement.

ACKNOWLEDGMENT

First, I would like to acknowledge and thank my dissertation advisor Dr. Jason T. L. Wang for his guidance, support and encouragement during the past several years. Without him, this work would have not been possible. I would also like to acknowledge and thank the members of my dissertation committee, Dr. James A. M. McHugh, Dr. David Nassimi, Dr. Christian E. Laing and Dr. Tamar Schlick. They have made lots of thoughtful suggestions to improve this work. I am deeply grateful for their guidance and supervision.

Moreover, I would like to thank all of the staff and graduate students in the Department of Computer Science whom I have known in the last several years. Special thanks also go to the members of the Data and Knowledge Engineering Laboratory. Each of these friends has aided me in numerous ways through these years at NJIT.

Finally, I would like to acknowledge and thank all members of my family, especially my wife, for their endless love, support and encouragement throughout my graduate study.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF TABLES
(Continued)

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Background

According to the well-known central dogma of molecular biology, RNA (Ribonucleic acid) is transcribed from DNA (Deoxyribonucleic acid) and plays a key role in the synthesis of proteins. Since the central dogma was first articulated in 1970, tRNA (transfer RNA) and mRNA (messenger RNA) have been extensively studied by molecular biologists. More recently, attention has been paid to non-coding RNAs (ncRNA). Many ncRNA genes have been discovered during the past decade.

There has been a great deal of effort in bioinformatics research on the development of sequence-based algorithms for RNA processing. However there has been relatively less work done in the area of RNA structure processing. Figure 1.1 depicts an example of an RNA secondary structure portrayed using the RNAplot of the Vienna RNA package [1]. In general, an RNA secondary structure includes stem-loops (hairpins), bulges, internal loops, multi-branch loops and pseudoknots. An RNA secondary structure is closely related to the function of its RNA molecule.

A sequence motif is a small segment of an RNA sequence that has a particular biological function. An RNA structural motif is a substructure of an RNA structure that has a particular biological function. Well-known RNA structural motifs include IRE (Iron Response Element) and HSL3 (Histone 3’ UTR stem-loop) [2,3]. As more and more RNA structural motifs were discovered, it became crucial to have a database holding these motifs for use in research. For example, Rfam [4] and RNA STRAND [5] are two databases of RNA structural motifs.

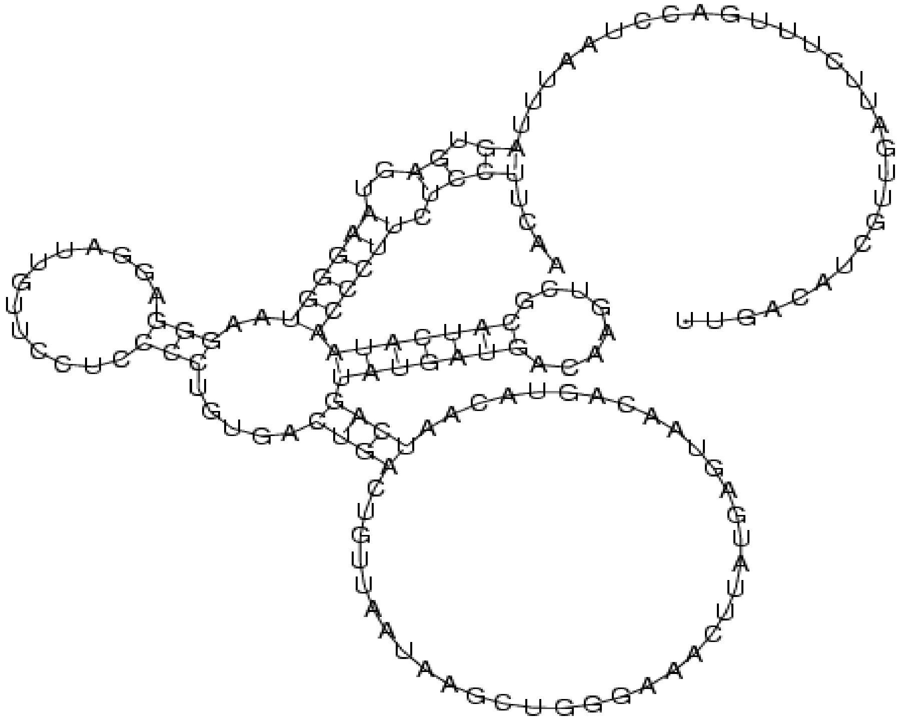


Figure 1.1 An example of an RNA secondary structure. This is QUAD RNA (Accession RF00113) of Rfam 9.0 [4].

As RNA structural motifs are archived, methods for matching, comparing and aligning a pair of RNA structural motifs become essential. With reference to RNA sequences, many tools have been designed for sequence matching and alignment. For instance, FASTA (FAST-All) [6] and BLAST (Basic Local Alignment Search Tool) [7] are examples of two outstanding software tools for sequence alignment. However, sequence level tools are not capable of matching, comparing and aligning RNA structures. In Figure 1.2, which shows a sequence logo diagram from the Weblogo tool [8], the two RNA sequences are identical on 12 out of 30 nucleotides, corresponding to a 40% similarity in the sequences. However, in terms of a comparison of their structures, they have a 100% similarity in structure. Several software tools have been developed for RNA structure alignment, such as RSmatch [9], Rsearch [10] and RNAforester [11]. In addition to sequence/structure matching and aligning, some software tools provide database searching as well, such as BLAST and RSmatch. They are able to accept a query motif from a user, based upon which they perform a motif search in a sequence/structure database.

```
>NM_000032
UUCGUUCGUCCUCAGUGCAGGGCAACAGGA
((((((.((((((......))))))))).)))
>NM_014585
CAACUUCAGCUACAGUGUUAGCUAAGUUUG
((((((.((((((......))))))))).)))
```

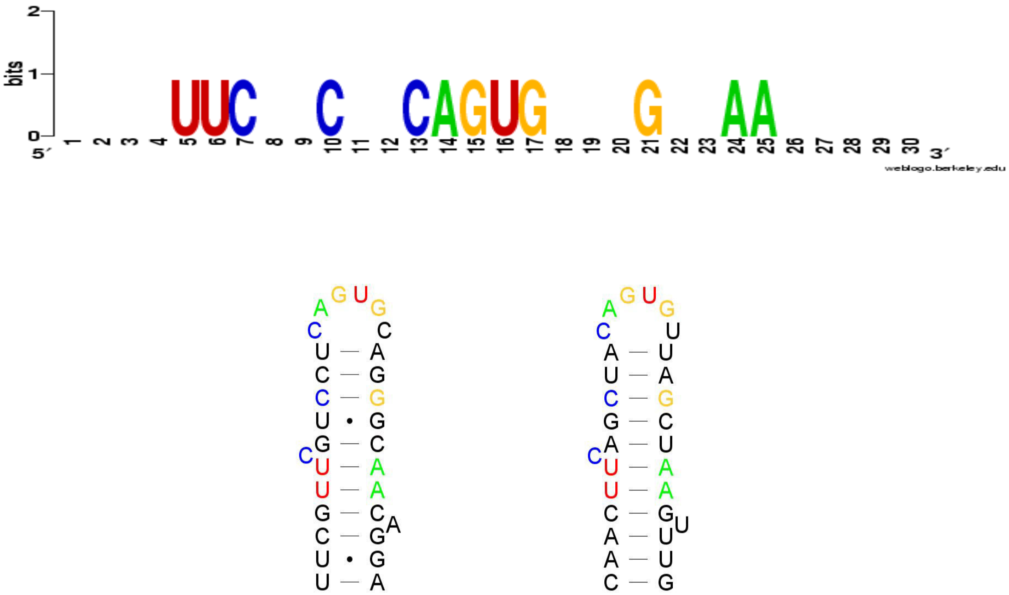


Figure 1.2 The comparison between sequence similarity and structure similarity in RNA molecules.

1.2 Motivation

Since the explosive expansion of the Internet and the World Wide Web in the late 1990's, web search engines for searching the Internet have become vital to both daily life and research. Speed and accuracy (in the sense of sensitivity and specificity) define the success of a web search engine. Without powerful, popular web search engines like Google, Yahoo! and Bing, the speed with which information could be acquired over the Internet would be slower by orders of magnitude.

Most RNA structure motif databases on the Internet only provide either keyword-based or sequence-based search methods, but lack structure-based search methods. A few software tools and RNA structure motif databases can perform off-line searches by allowing users to download to their local machines, an approach which is inconvenient for most users. In addition, none of the RNA structure motif databases on the Internet offer fast and accurate structure-based searches.

Furthermore, none of the online web servers is able to search and predict RNA tertiary motifs. Since the advanced improvement of the crystallography on the RNA molecules in recent years, the study, analysis and prediction on the RNA tertiary structures has become extensive; this had not been possible for the past decades.

Therefore, in this dissertation, a cyberinfrastructure, named RNAcyber, capable of performing RNA motif search and prediction, is proposed, designed and implemented. As part of RNAcyber, a comprehensive study of how to build a fast, high-recall and high-precision structure-based search engine for RNA motif databases was carried out. As another part of RNAcyber, web servers, which are capable of detecting RNA secondary structure motifs and predicting RNA tertiary motifs from the RNA secondary structure level, has been developed.

1.3 Organization of the Dissertation

In Chapter 2, the first component of RNAcyber is introduced. It is a web-based search engine named RmotifDB. This web-based tool integrates an RNA secondary structure comparison algorithm with the secondary structure motifs stored in the Rfam database. With a user-friendly interface, RmotifDB provides the ability to search for ncRNA structure motifs by both structural and sequential methods. The second component of RNAcyber is an enhanced version of RmotifDB, which is introduced in Chapter 3. This enhanced version combines data from multiple sources, incorporates a variety of well-established structure-based search methods, and is integrated with the Gene Ontology. To display RmotifDB's search results, a software tool, called RSview, is developed. RSview is able to display the search results in a graphical manner, which is described in Chapter 4.

In Chapters 5 and 6, RNAcyber contains a web-based tool called Junction-Explorer, which employs a data mining method for predicting tertiary motifs in RNA junctions. The classifier of Junction-Explorer is trained on solved RNA tertiary structures obtained from the Protein Data Bank [12], and is able to predict the configuration of coaxial helical stacks and families (topologies) in RNA junctions.

Finally, the contributions and conclusions of this dissertation, as well as future work, are presented in Chapter 7.

CHAPTER 2

A SIMPLE RNA STRUCTURE SEARCH ENGINE AND DATABASE

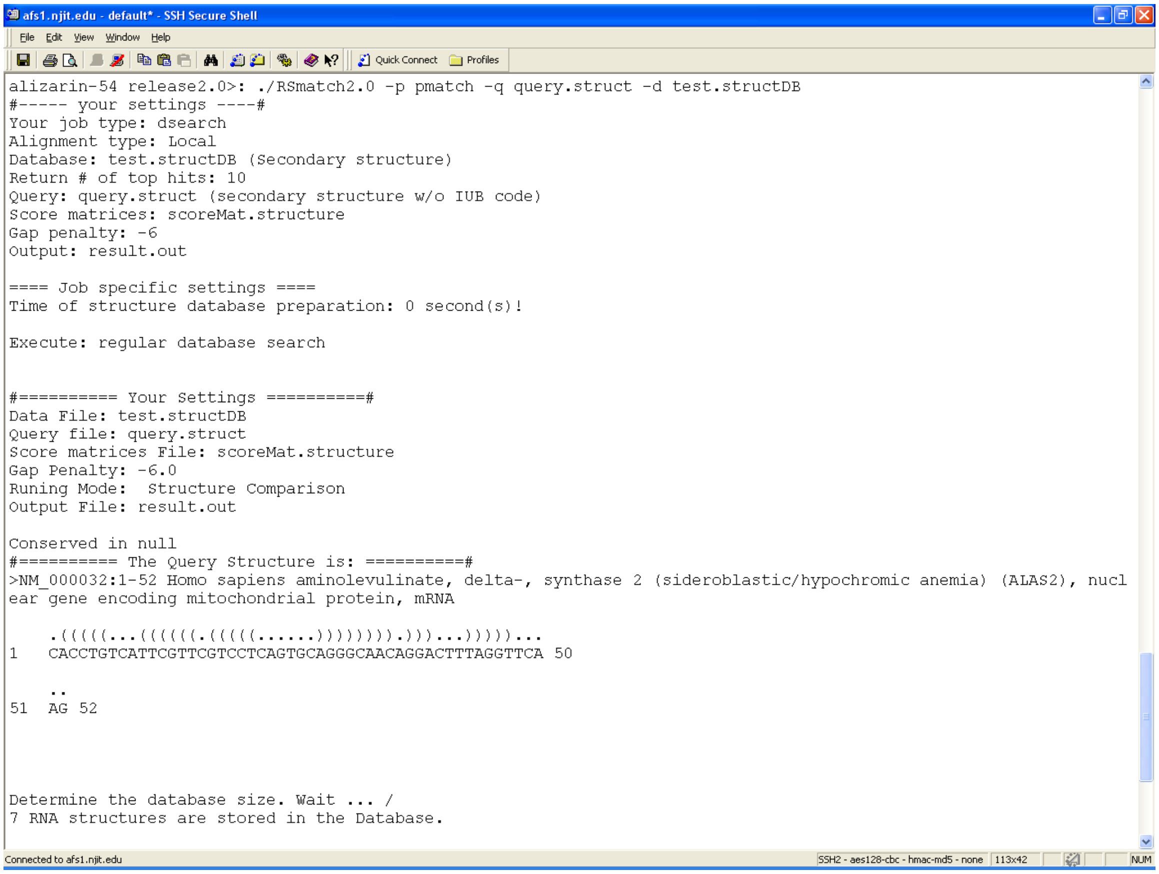
2.1 Preface

In this chapter, a simple RNA structure search engine with its own database named RmotifDB 1.0 is presented. RSmatch [9] is used as the core of the search engine in RmotifDB 1.0. The RNA structure motifs deposited in the database of RmotifDB 1.0 are extracted from Rfam [4]. In the following section, RSmatch, Rfam and the detailed design of RmotifDB 1.0 are presented.

2.2 RSmatch

RSmatch is a software tool for comparing two RNA structures and for RNA motif detection. It is intended to offer a light-weight approach to the comparison of RNA structures. RSmatch is used as the core of the RmotifDB 1.0 search engine. RSmatch is fast, taking quadratic time as determined by the size of the two given RNA structures. Specifically, its time complexity is *O(mn)* where *m* is the length of the query RNA structure and *n* is the length of the subject RNA structure.

Functional RNA motifs can be usefully studied by aligning RNA secondary structures. Recently, many software tools have been developed to find RNA motifs by aligning RNA structures. However, existing software tools have two major drawbacks. First, they require a large number of pre-aligned structures. Secondly, they have high time complexities. Therefore, these tools have difficulty in processing RNAs without pre-aligned structures and in handling large RNA structure databases.


afs1.njit.edu - default* - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles
Connected to afs1.njit.edu
SSH2 - aes128-cbc - hmac-md5 - none
113x42
NUM

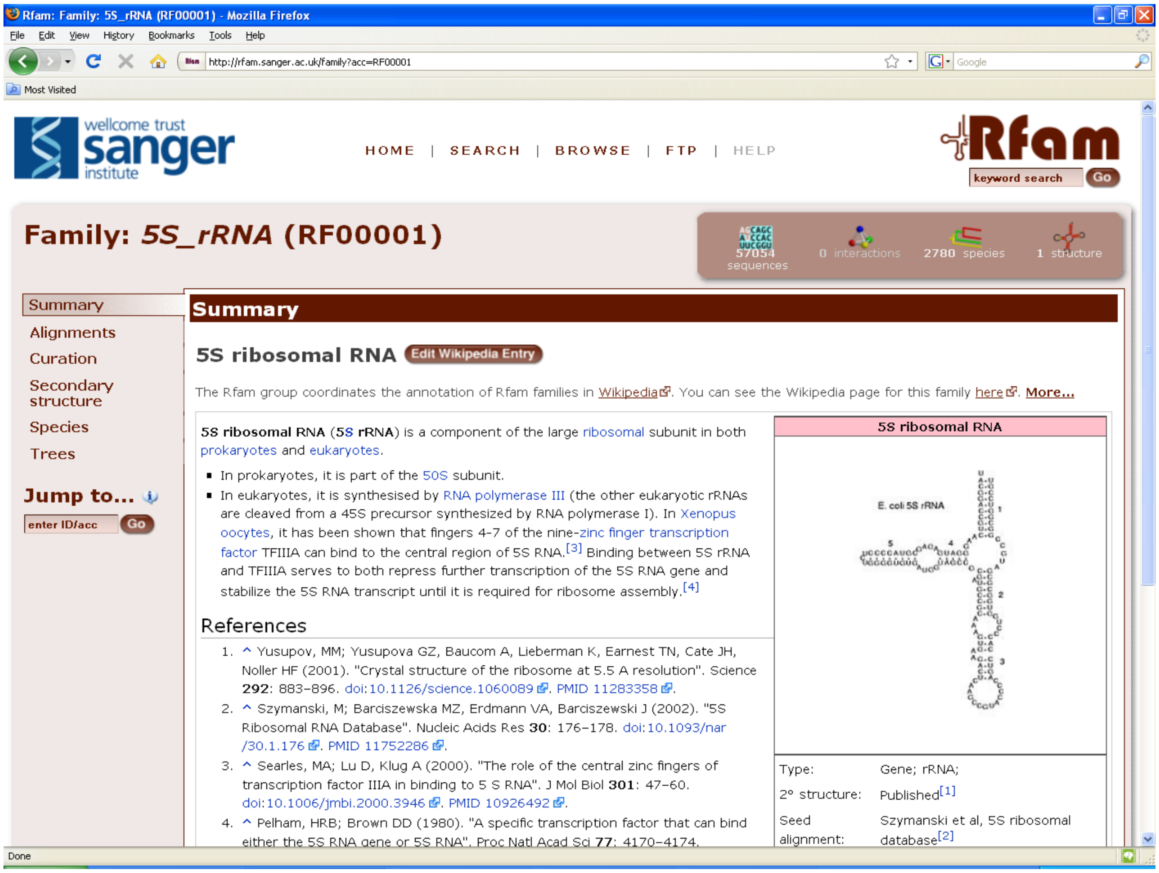
```
alizarin-54 release2.0>: ./RSmatch2.0 -p pmatch -q query.struct -d test.structDB
#----- your settings ----#
Your job type: dsearch
Alignment type: Local
Database: test.structDB (Secondary structure)
Return # of top hits: 10
Query: query.struct (secondary structure w/o IUB code)
Score matrices: scoreMat.structure
Gap penalty: -6
Output: result.out

==== Job specific settings ====
Time of structure database preparation: 0 second(s)!

Execute: regular database search


#========== Your Settings ==========#
Data File: test.structDB
Query file: query.struct
Score matrices File: scoreMat.structure
Gap Penalty: -6.0
Runing Mode:  Structure Comparison
Output File: result.out

Conserved in null
#========== The Query Structure is: ==========#
>NM_000032:1-52 Homo sapiens aminolevulinate, delta-, synthase 2 (sideroblastic/hypochromic anemia) (ALAS2), nucl
ear gene encoding mitochondrial protein, mRNA

    .(((((...((((((.(((((......)))))))).)))...)))))...
1   CACCTGTCATTCGTTCGTCCTCAGTGCAGGGCAACAGGACTTTAGGTTCA 50

    ..
51  AG 52



Determine the database size. Wait ... /
7 RNA structures are stored in the Database.
```

Figure 2.1 The execution of RSmatch under the Unix command line environment.

RSmatch is an efficient tool for RNA motif detection and the alignment of RNA secondary structures. Its algorithm decomposes an RNA secondary structure into a collection of non-decomposable structure components. In order to capture the structural particularities, RSmatch uses a tree model to organize these structure components.

RSmatch aligns a pair of RNA secondary structures using two separate scoring matrices that operate in both a local and global manner. One scoring matrix is used for single-stranded regions and the other is used for double-stranded regions. Furthermore, when searching an RNA structure database, RSmatch can detect similar RNA substructures and perform iterative database searches and multiple structure alignments. This establishes that RSmatch is able to identify functional RNA structural motifs.

By conducting experiments with instances of known RNA structure motifs, including simple stem-loops and complex structures with junctions, it has been demonstrated that the accuracy of RSmatch is outstanding when compared to other software tools [9]. It is currently the leader among software tools for structural alignment in terms of computing efficiency and accuracy. RSmatch is especially useful to scientists and researchers interested in aligning RNA structural motifs from RNA folding programs or wet lab experiments where the size of the RNA structure dataset is very large. The software is available for download from http://datalab.njit.edu/biodata/rna/RSmatch/software.htm. Figure 2.1 presents a screenshot of the execution of RSmatch's structural database search under the Unix command line environment.



Family: *5S_rRNA* (RF00001)

Summary

5S ribosomal RNA

The Rfam group coordinates the annotation of Rfam families in Wikipedia. You can see the Wikipedia page for this family here. More...

5S ribosomal RNA (**5S rRNA**) is a component of the large ribosomal subunit in both prokaryotes and eukaryotes.

- In prokaryotes, it is part of the 50S subunit.
- In eukaryotes, it is synthesised by RNA polymerase III (the other eukaryotic rRNAs are cleaved from a 45S precursor synthesized by RNA polymerase I). In Xenopus oocytes, it has been shown that fingers 4-7 of the nine-zinc finger transcription factor TFIIIA can bind to the central region of 5S RNA.[3] Binding between 5S rRNA and TFIIIA serves to both repress further transcription of the 5S RNA gene and stabilize the 5S RNA transcript until it is required for ribosome assembly.[4]

References

1. ^ Yusupov, MM; Yusupova GZ, Baucom A, Lieberman K, Earnest TN, Cate JH, Noller HF (2001). "Crystal structure of the ribosome at 5.5 A resolution". Science **292**: 883–896. doi:10.1126/science.1060089. PMID 11283358.
2. ^ Szymanski, M; Barciszewska MZ, Erdmann VA, Barciszewski J (2002). "5S Ribosomal RNA Database". Nucleic Acids Res **30**: 176–178. doi:10.1093/nar/30.1.176. PMID 11752286.
3. ^ Searles, MA; Lu D, Klug A (2000). "The role of the central zinc fingers of transcription factor IIIA in binding to 5 S RNA". J Mol Biol **301**: 47–60. doi:10.1006/jmbi.2000.3946. PMID 10926492.
4. ^ Pelham, HRB; Brown DD (1980). "A specific transcription factor that can bind either the 5S RNA gene or 5S RNA". Proc Natl Acad Sci **77**: 4170–4174.

5S ribosomal RNA	
Type:	Gene; rRNA;
2° structure:	Published[1]
Seed alignment:	Szymanski et al, 5S ribosomal database[2]

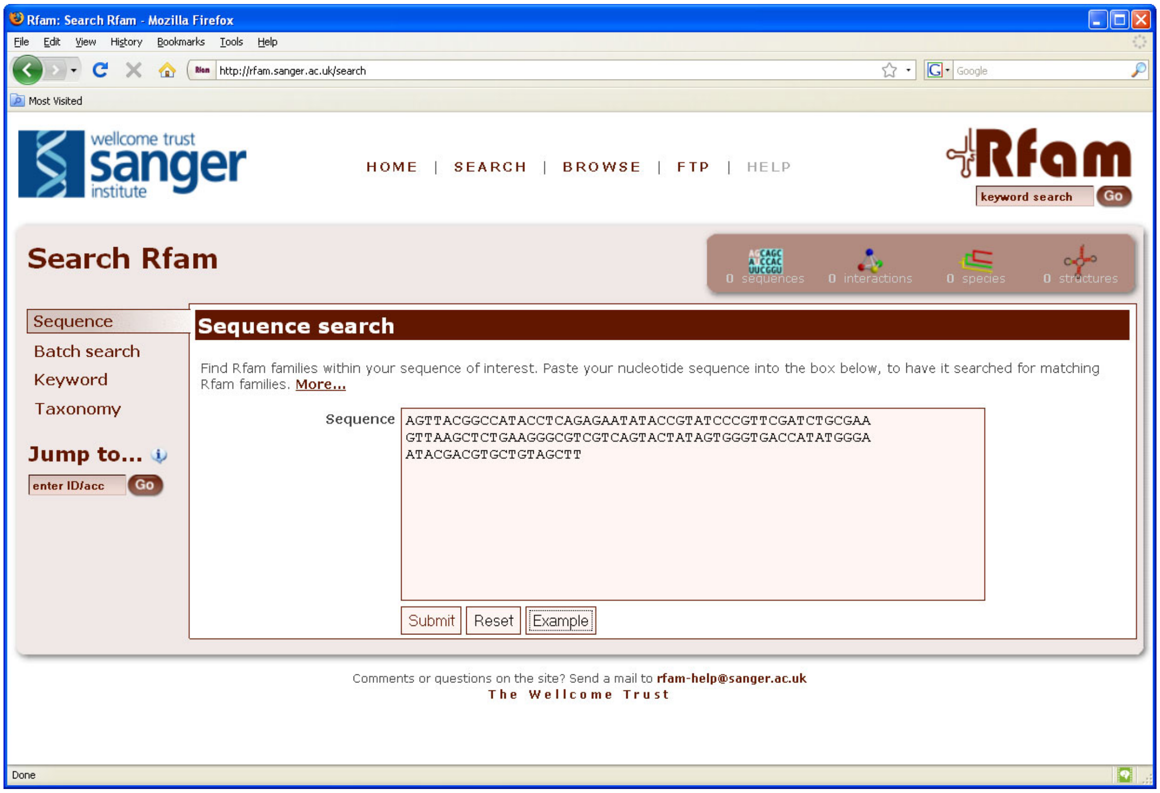
Figure 2.2 The entry page of one ncRNA family (5S ribosomal RNA) in the Rfam 9.0 database [4].

2.3 Rfam Database

Rfam is a well-annotated, open access database which is a depository for information on non-coding RNA (ncRNA) families and other RNA structural motifs. Rfam collects covariance models and multiple sequence alignments which are used to represent non-coding RNA families. The latest version of Rfam 9.0, containing a total of 603 families, is available at http://rfam.sanger.ac.uk/. Figure 2.2 shows the entry page for one ncRNA family (5S ribosomal RNA).

By giving a query sequence, the user can search the entire 603 sets of the covariance models representing the non-coding RNA families. Since the cost of computation using a covariance model is very expensive, an initial BLAST search is performed to decrease the size of the search space. When a search is completed, the search results are displayed in the browser and list the RNA families which have a distinct similarity to the input query sequence. The user can view the multiple sequence alignments and the annotation of RNA families listed on the search result. The interface for sequence search in the Rfam database is shown in Figure 2.3. In addition to a search based on query sequence, the Rfam website allows the user to search ncRNA families based on keyword and taxonomy characteristics. The Rfam database can be downloaded in plain text format from the Rfam website and searched offline using the Infernal package [13] on user's local machine.

The secondary structures of ncRNAs may be similar without similarity in their underlying sequential. Therefore, multiple sequence alignments with additional secondary structure information for these ncRNA families may provide a useful way to allow users to study ncRNA function and structure.


Rfam: Search Rfam - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://rfam.sanger.ac.uk/search
Google
Most Visited
wellcome trust sanger institute
HOME | SEARCH | BROWSE | FTP | HELP
Rfam
keyword search
Go
0 sequences
0 interactions
0 species
0 structures
Sequence
Batch search
Keyword
Taxonomy
Jump to...
enter ID/acc
Go
Submit
Reset
Example
Done

Search Rfam

Sequence search

Find Rfam families within your sequence of interest. Paste your nucleotide sequence into the box below, to have it searched for matching Rfam families. More...

Sequence

```
AGTTACGGCCATACCTCAGAGAATATACCGTATCCCGTTCGATCTGCGAA
GTTAAGCTCTGAAGGGCGTCGTCAGTACTATAGTGGGTGACCATATGGGA
ATACGACGTGCTGTAGCTT
```

Comments or questions on the site? Send a mail to rfam-help@sanger.ac.uk
The Wellcome Trust

Figure 2.3 The interface for Rfam's sequence search method [4].

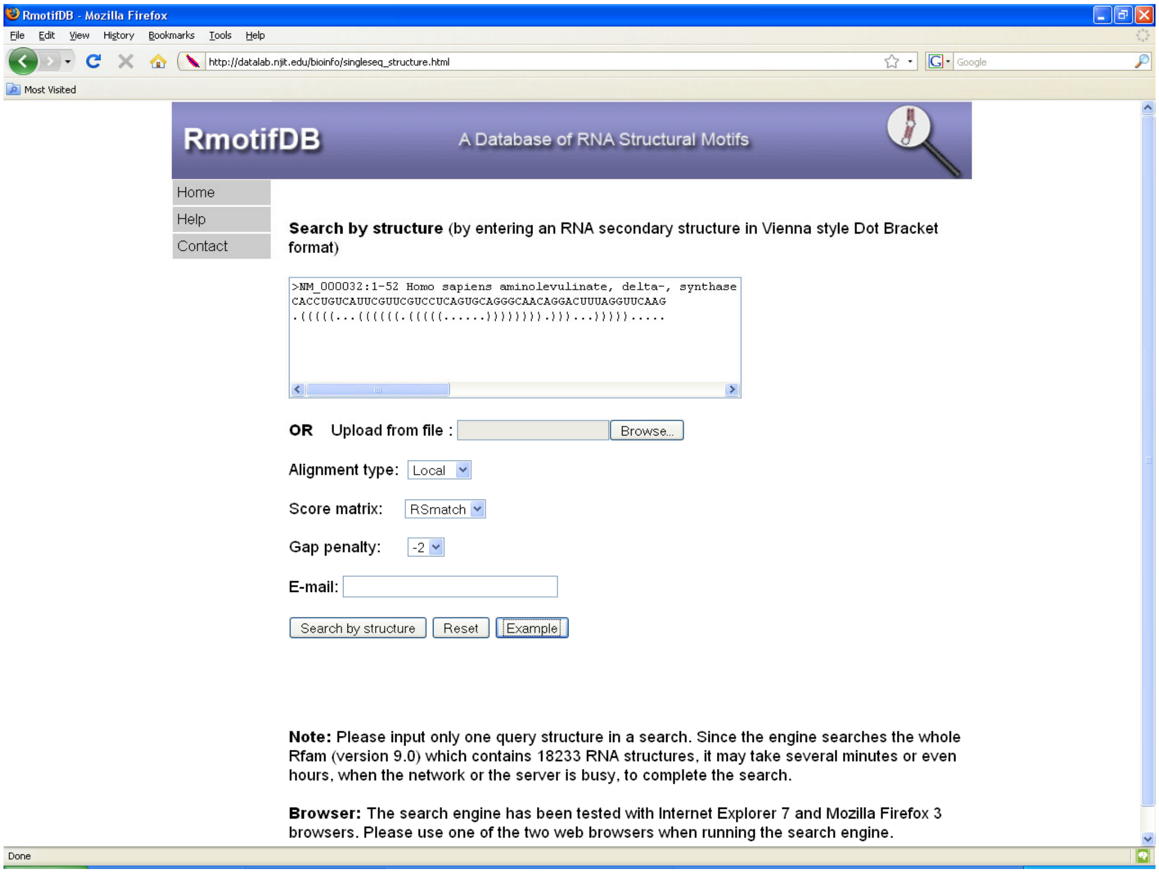
In the Rfam database, the multiple sequence alignments represent information on the secondary structure and sequences of ncRNA families. Moreover, the multiple sequence alignments can be transferred to a statistical model using so-called profile stochastic context-free grammars (SCFGs). This is also known as the covariance model and is very similar to the hidden Markov models used in the Pfam database for the protein family annotation.

In Rfam, one SCFG and two multiple sequence alignments are used to represent each ncRNA family. The first multiple sequence alignment is called the seed alignment. The second alignment is called the full alignment. The seed alignment, which is generated manually by biological experiment, includes representative members of the ncRNA family and is annotated with secondary structural information. The seed alignment is also used to generate the SCFG or covariance model (CM) by utilizing the Infernal package which is used to detect new family members and add them to the alignment of the family. The expanded alignment generated by computation (as opposed to manually) including the newly added family members found by the Infernal package is called the full alignment. The newly detected family members are added to both the alignment and the covariance model. The full alignment is thus the result of a search that uses SCFG against the sequence database via the Infernal package. The initial seed alignment is also retained because of its special biological status or pedigree.

2.4 The RmotifDB 1.0 System

RSmatch offers an efficient algorithm for aligning two RNA structures, along with a basic RNA database search capability. However it must be run offline on a user's local machine, which is a major drawback. Even RADAR (http://datalab.njit.edu/biodata/rna/RSmatch/server.htm) [14], a descendant of RSmatch with an excellent web interface for aligning two RNA structures, does not contain a search engine function for a large database. In the previous section, it has been observed that there are provisions for sequence, keyword and taxonomy searching in Rfam database, but not for structure searching. This underscores the fact that to intensively study RNA structural functions or motifs, a structure-based search engine for RNA motif databases is needed. With this motivation, RmotifDB 1.0 was built, the first prototype of this study, available at http://datalab.njit.edu/bioinfo/singleseq_index.html.

RmotifDB 1.0 supports searching for the “nearest neighbors” of RNA structural motifs from its database. The nearest neighbors of an RNA structural motif are other motifs with a high degree of similarity to the given motif. There are currently 18,233 RNA structures from the combined 603 Rfam family (version 9.0) seed alignments deposited in RmotifDB 1.0 database. RSmatch version 2.0 is used as the core of the search engine for RmotifDB 1.0. The two major search modes are provided as search-by-sequence and search-by-structure. On completion of a search, an email notification is sent to the user. Since the search engine accesses the whole Rfam (version 9.0) with its 18,000 plus RNA structures, it may take minutes or even hours to complete the search when the server is busy.


RmotifDB - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://datalab.njit.edu/bioinfo/singleseq_structure.html
Google
Most Visited
RmotifDB
A Database of RNA Structural Motifs
Home
Help
Contact
Browse...
Search by structure
Reset
Example
Done

Search by structure (by entering an RNA secondary structure in Vienna style Dot Bracket format)

```
>NM_000032:1-52 Homo sapiens aminolevulinate, delta-, synthase
CACCUGUCAUUCGUUCGUCCUCAGUGCAGGGCAACAGGACUUUAGGUUCAAG
.(((((...((((((.(((((......)))))))).)))...))))).....
```

OR Upload from file :

Alignment type: Local

Score matrix: RSmatch

Gap penalty: -2

E-mail:

Note: Please input only one query structure in a search. Since the engine searches the whole Rfam (version 9.0) which contains 18233 RNA structures, it may take several minutes or even hours, when the network or the server is busy, to complete the search.

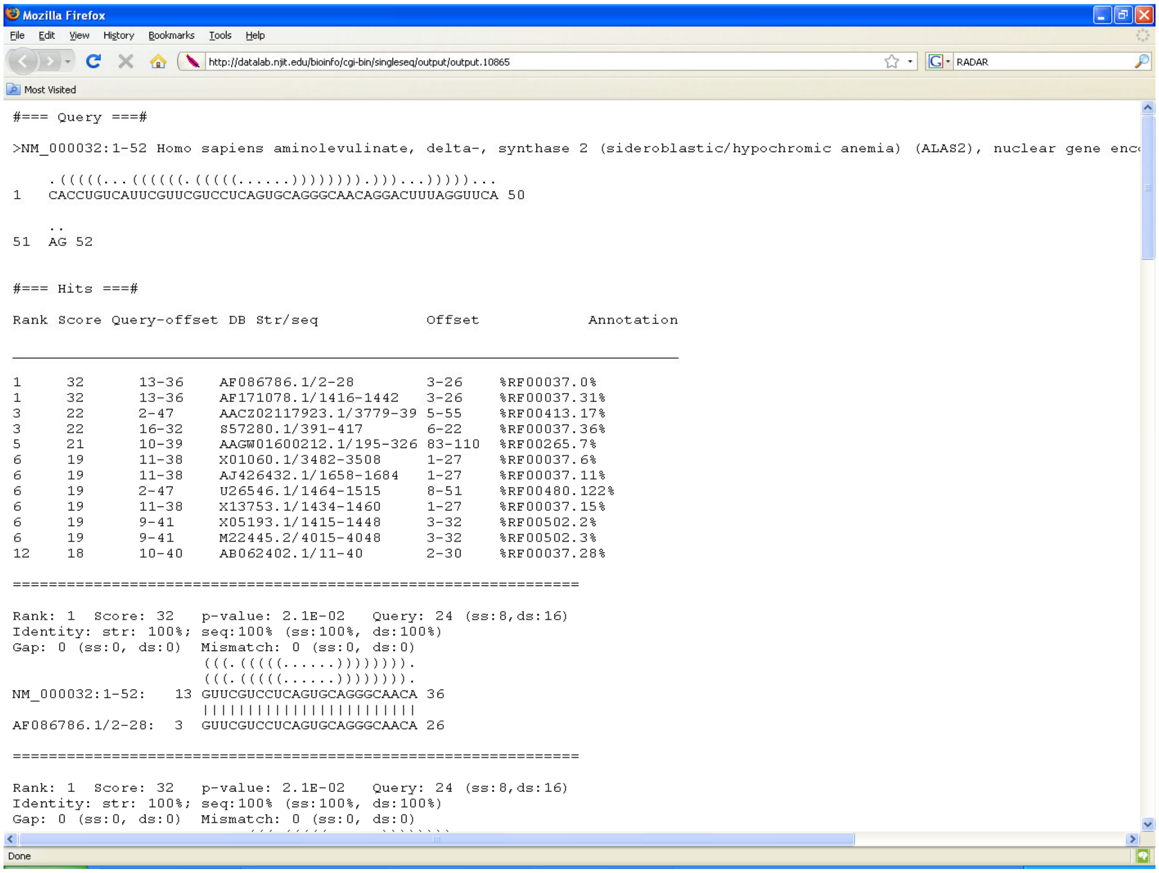
Browser: The search engine has been tested with Internet Explorer 7 and Mozilla Firefox 3 browsers. Please use one of the two web browsers when running the search engine.

Figure 2.4 Screenshot of RmotifDB 1.0 with search by structure function.

In order to build the database for RmotifDB 1.0, the plain text seed alignment file with 603 ncRNA families is downloaded from the Rfam 9.0 website. A total of 18,233 ncRNA sequences are extracted from this seed alignment file. Each of these sequences is then folded, using the Vienna RNA package's RNAfold [1] to obtain their structures. Finally, the entire group of 18,233 ncRNA sequences along with their structure information is stored in a single plain text file which constitutes the major database file for RmotifDB 1.0.

RSmatch 2.0 is used as the search engine for RmotifDB 1.0. The RSmatch 2.0 software is downloaded from the RADAR website. The user's query RNA structure and the major database file of RmotifDB 1.0 are the two input files for RSmatch. RSmatch generates a search report with a ranked list for the user query RNA structure against the RmotifDB 1.0 database file. The implementation uses a perl-cgi approach to integrate the web interface with RSmatch. This allows use of the search engine over the web via a browser. Figure 2.4 illustrates the web interface for search-by-structure for RmotifDB 1.0.

With an improved web interface, the user can submit the query input which is given as either an RNA sequence or an RNA structure. If the RNA sequence is given, it must be in FASTA format [6]. If the RNA structure is given, it must be in Vienna dot-bracket format [1]. The user can either paste the input query into a text box or upload the query input from a plain text file. Additional options include variations on the alignment type, the score matrix, and the gap penalty. Local or global alignment can be selected as the alignment type. Currently, the only score matrix used is RSmatch's default matrix, but additional options for score matrices can be accommodated. The default gap penalty is -2, which can be changed, based on the user's preference. The user's e-mail address is required and used for notification once the search results are available.



http://datalab.njit.edu/bioinfo/cgi-bin/singleseq/output/output.10865

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

>NM_000032:1-52 Homo sapiens aminolevulinate, delta-, synthase 2 (sideroblastic/hypochromic anemia) (ALAS2), nuclear gene enc

    .(((((...((((((.(((((......)))))))).)))...)))))...
1   CACCUGUCAUUCGUUCGUCCUCAGUGCAGGGCAACAGGACUUUAGGUUCA 50

    ..
51  AG 52


#=== Hits ===#

Rank Score Query-offset DB Str/seq              Offset          Annotation

___________________________________________________________________________

1     32      13-36     AF086786.1/2-28         3-26    %RF00037.0%
1     32      13-36     AF171078.1/1416-1442    3-26    %RF00037.31%
3     22      2-47      AACZ02117923.1/3779-39  5-55    %RF00413.17%
3     22      16-32     S57280.1/391-417        6-22    %RF00037.36%
5     21      10-39     AAGW01600212.1/195-326  83-110  %RF00265.7%
6     19      11-38     X01060.1/3482-3508      1-27    %RF00037.6%
6     19      11-38     AJ426432.1/1658-1684    1-27    %RF00037.11%
6     19      2-47      U26546.1/1464-1515      8-51    %RF00480.122%
6     19      11-38     X13753.1/1434-1460      1-27    %RF00037.15%
6     19      9-41      X05193.1/1415-1448      3-32    %RF00502.2%
6     19      9-41      M22445.2/4015-4048      3-32    %RF00502.3%
12    18      10-40     AB062402.1/11-40        2-30    %RF00037.28%

================================================================

Rank: 1  Score: 32   p-value: 2.1E-02   Query: 24 (ss:8,ds:16)
Identity: str: 100%; seq:100% (ss:100%, ds:100%)
Gap: 0 (ss:0, ds:0)  Mismatch: 0 (ss:0, ds:0)
                     (((.(((((......)))))))).
                     (((.(((((......)))))))).
NM_000032:1-52:   13 GUUCGUCCUCAGUGCAGGGCAACA 36
                     ||||||||||||||||||||||||
AF086786.1/2-28:  3  GUUCGUCCUCAGUGCAGGGCAACA 26

================================================================

Rank: 1  Score: 32   p-value: 2.1E-02   Query: 24 (ss:8,ds:16)
Identity: str: 100%; seq:100% (ss:100%, ds:100%)
Gap: 0 (ss:0, ds:0)  Mismatch: 0 (ss:0, ds:0)
```

Figure 2.5 Screenshot of a search report generated by RmotifDB 1.0.

Upon completion of the search (which, as previously observed, may take minutes or hours), an email notification is sent with a link to the results. Figure 2.5 illustrates a search result for RmotifDB 1.0.

RmotifDB 1.0 capitalized on RSmatch and the Rfam database to build a basic search engine and database. By presenting a convenient browser style interface, RmotifDB 1.0 provides the ability to search for ncRNA structural motifs in both structural and sequential ways, benefiting those researchers interested in ncRNA's structural functions and structural motifs. In the next chapter, RmotifDB 2.0, an enhanced version of RmotifDB 1.0, is presented, which is further enhanced with an improved search engine function and internal database.

CHAPTER 3

AN INTEGRATED RNA STRUCTURE SEARCH ENGINE AND DATABASE

3.1 Preface

In this chapter, the design and implementation of an advanced RNA structural motif database RmotifDB 2.0 is presented. The RNA structural motifs stored in RmotifDB 2.0 derive from those

- Collected manually from the biomedical literature,
- Submitted by scientists from around the world, or
- Discovered using a variety of motif mining methods.

A motif mining method is described in detail. The interface and search mechanisms provided by RmotifDB 2.0 is also presented as well as techniques used to integrate RmotifDB 2.0 with Gene Ontology. The RmotifDB 2.0 system is fully operational and available at http://datalab.njit.edu/bioinfo/UTRdb/.

3.2 RNA Structural Motifs

Post-transcriptional control is one of the mechanisms that regulate gene expression in eukaryotic cells. RNA elements residing in the UnTranslated Regions (UTRs) of mRNAs have been shown to play a variety of roles in post-transcriptional control, including mRNA localization, translation, and stability [3,15]. The RNA elements in UTRs can be roughly divided into three categories: elements whose functions are primarily attributable to their sequences, elements whose functions are attributable to their secondary or tertiary structures, and elements whose functions are attributable to both their sequences and structures. For simplicity, the first category is called sequence elements, and the second and third are called structure elements (or structural motifs), respectively.

Well-known sequence elements include AU-rich elements (AREs), which contain one or several tandem AUUUA sequences and are involved in regulating mRNA stability [16], and miRNA target sequences, which are partially complementary to cognate miRNA sequences and are involved in regulating translation or mRNA stability [17].

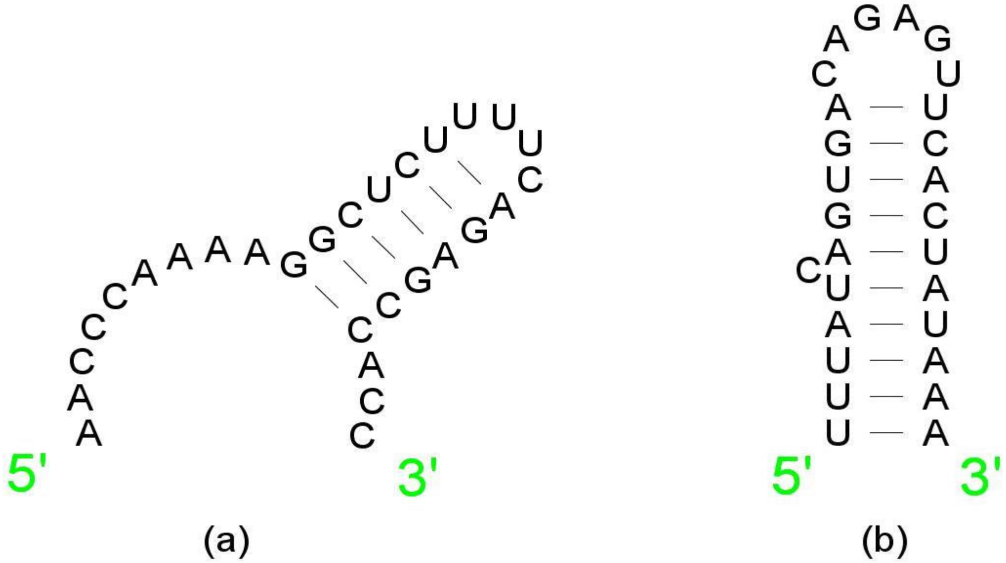


Figure 3.1 (a) An example of the HSL3 motif. (b) An example of the IRE motif.

Well-known structure elements (or structural motifs) include the histone 3'-UTR stem-loop structure (HSL3) and the iron response element (IRE) [2,3]. Both sequence and structure are important to the functions of the structural motifs. HSL3 is a stem-loop structure of about 25 nucleotides that exists in the 3'-UTRs of most histone genes. Figure 3.1a portrays an HSL3 motif using the XRNA tool (http://rna.ucsc.edu/rnacenter/

xrna/xrna.html). The HSL3 structure is critical for both termination of the transcription of mRNAs and the stability of mRNAs. These functions are exerted by the stem-loop binding protein (SLBP) that interacts with HSL3. IRE is a stem-loop structure of about 30 nucleotides with a bulge or a small internal loop in the stem (Figure 3.1b). 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 to the iron regulatory proteins (IRPs) of those species which control the translation and stability of IRE-containing mRNAs.

HSL3 and IRE have several similarities: both are small simple RNA structures with less than 40 nucleotides; both exist in the UTRs of several genes with related functions; and both bind to cellular proteins and are involved in post-transcriptional gene regulation. These regulations via HSL3 and IRE constitute a distinct mode of gene regulation whereby the expression of several genes can be modulated via a common RNA structure in UTRs. Functional sequence motifs in genomes have been extensively studied in recent years, particularly for the promoter region and sequences involved in splicing [18-20]. By contrast, RNA structure elements have been investigated to a much lesser extent, largely due to the difficulties involved in predicting correct RNA structures and in conducting RNA structure alignments which have entailed huge computing costs.

Some success has been achieved in making accurate RNA structure prediction using phylogenetic approaches [21] and sequence alignments [22,23]. However, large-scale mining for conserved structures in eukaryotic UTRs has been studied to a lesser extent. Furthermore, current methods for finding common stem-loop structures rely solely on the detection of structural similarities [24]. Gene Ontology information has not been used in the study of RNA structure, although integrating ontologies with other biological data has been studied extensively [25-29].

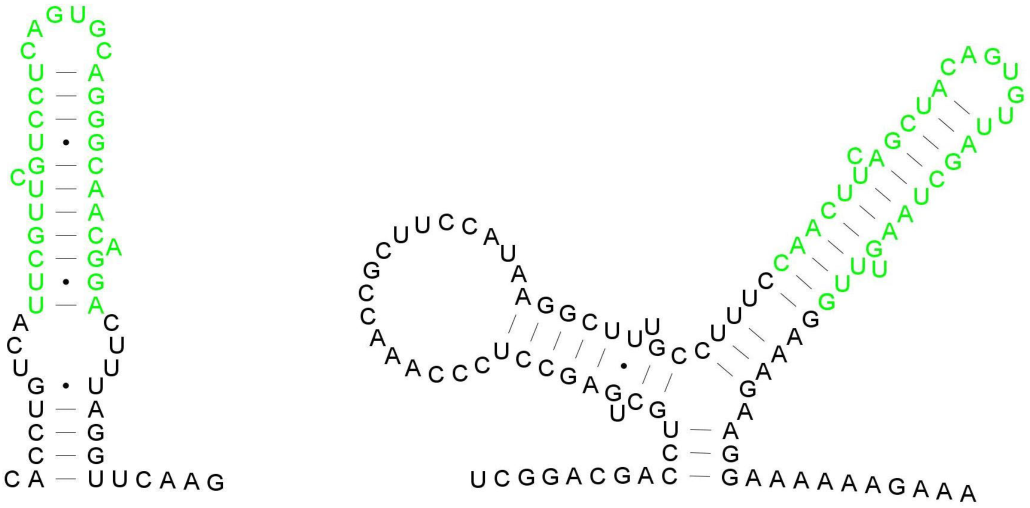


Figure 3.2 Alignment of two RNA secondary structures where the local matches found by RSmatch are highlighted with the color green.

Here is presenting an improved version of the search engine and database from Chapter 2, RmotifDB 2.0, that contains structural motifs found in 5' and 3' UTRs of eukaryotic mRNAs. The RNA structural motifs are linked with Gene Ontology and PubMed entries relevant to the motifs. A wide variety of motif mining methods are developed. In particular, in Section 3.3, a histogram-based method for discovering motifs in eukaryotic UTRs is presented and the detail of the histogram-based method is described. In Section 3.4, RmotifDB 2.0 is presented, as well as its interface and search mechanisms. In Section 3.5, techniques used to integrate RmotifDB 2.0 with Gene Ontology is described. Section 3.6 summarizes the conclusions and indicates some possibilities for future research.

3.3 A Motif Mining Method

Several structural motif mining methods based on different RNA representation models have been developed. For example, the work [30-32] represented an RNA secondary structure using an ordered labeled tree, and designed a tree matching algorithm to find motifs in multiple RNA secondary structures. As described in Chapter 2, RSmatch uses a loop model for representing RNA secondary structures. In RSmatch, a dynamic programming algorithm for aligning a pair of RNA secondary structures based on this loop model was utilized. The time complexity of RSmatch is $O(mn)$, where m and n are the sizes (number of nucleotides) of the two compared secondary structures. RSmatch is available at the RADAR server (acronym for RNA Data Analysis and Research) [14] accessible at http://datalab.njit.edu/biodata/rna/RSmatch/server.htm. Figure 3.2 shows the common region of two RNA secondary structures for homo sapiens sequences portrayed using XRNA. The local matches found by RSmatch are highlighted in green.

A histogram-based scoring method is described below, that is for discovering novel conserved RNA stem-loops in eukaryotic UTRs using RSmatch. This method is an extension of a previously developed histogram-based algorithm for DNA sequence classification [33]. Given a set of RNA secondary structures, the method uses RSmatch to perform pairwise alignments by comparing two RNA structures from the set at a time. Given an optimal local alignment between two structures A and B found by RSmatch, the set of bases in the aligned region of A is denoted by $Q_A = \{A_i, A_{i+1}, ..., A_j\}$ where A_i (A_j) is the 5'-most (3'-most,) nucleotide not aligned to a gap. The set of bases in the aligned region of B is denoted by $Q_B = \{B_m, B_{m+1}, ..., B_n\}$ where B_m (B_n) corresponds to the 5'-most (3'-most) nucleotide not aligned to a gap. Each nucleotide $A_k \in Q_A$ that is not aligned to a gap scores | j-i+1 | points. All other bases in the structure A receive zero points. Thus, the larger the aligned region between A and B, the higher the score each base in the region receives. When aligning the structure A with another structure C, some bases in Q_A may receive non-zero points, hence the scores of those bases are accumulated. Therefore, the bases in a conserved RNA motif will have high scores.

To validate this approach, experiments is conducted to evaluate the effectiveness of this scoring method. The conserved stem-loops considered were IRE motifs containing about 30 nucleotides, located in the 5'-UTRs or 3'-UTRs of mRNAs coding for proteins involved in cellular iron metabolism. The test dataset was prepared as follows. By searching human RefSeq mRNA sequences from NCBI (the National Center for Biotechnology Information at http://www.ncbi.nlm.nih.gov/RefSeq/), several mRNA sequences were obtained within each of which at least one IRE motif was known to exist. Then the sequences' UTR regions were extracted as indicated by RefSeq's GenBank annotation and used PatSearch [34] to locate the IRE sequences. Each IRE sequence was then extended from both ends to obtain 100 nucleotide sequences. These sequences were mixed with several "noisy" sequences of the same length. The resulting sequences were then folded using the Vienna RNA package [1], with the package's RNAsubopt function assigned a setting of "-e 0". It is noted that this setting may yield multiple RNA structures with the same free energy for any given RNA sequence.

Figure 3.3 shows the score histograms for three tested RNA structures. Clusters of bases with high scores correspond to the IRE motifs in the RNA structures. Similar clusters of bases with high scores corresponding to the IRE motifs were observed in the other IRE-containing RNA structures, but not in the "noisy" structures. This result indicates that this histogram-based scoring method is able to detect biologically significant motifs in multiple RNA structures.

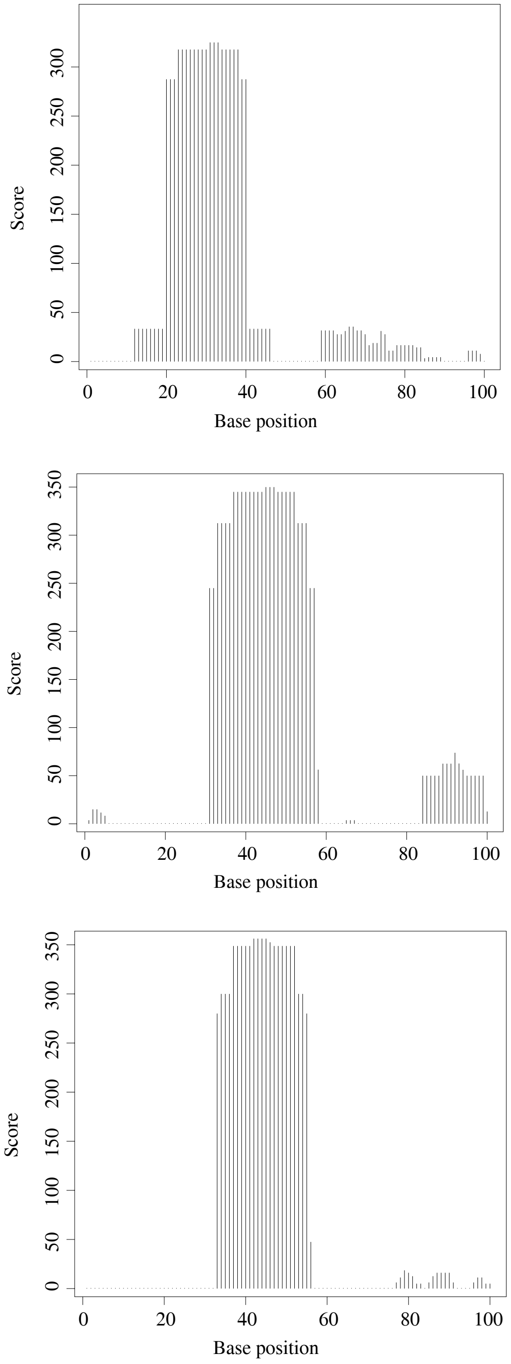
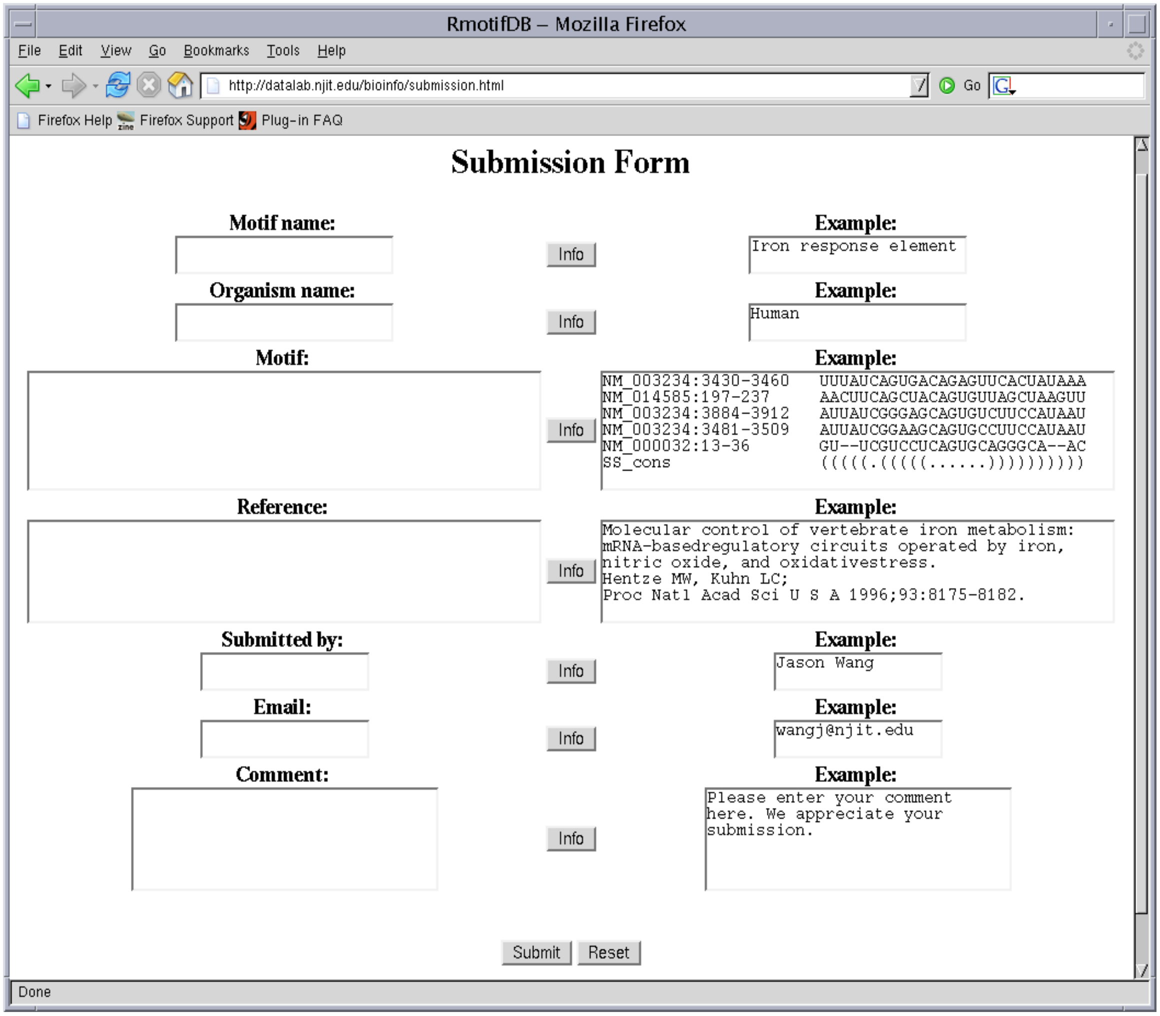


Figure 3.3 Diagrams illustrating the effectiveness of the proposed scoring method. IRE is found around base positions 20-60 in the RNA structures corresponding to the respective diagrams.


RmotifDB – Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Go
Firefox Help
Firefox Support
Plug-in FAQ
Info
Submit
Reset
Done

http://datalab.njit.edu/bioinfo/submission.html

Submission Form

Motif name: | **Example:** Iron response element

Organism name: | **Example:** Human

Motif: | **Example:**

```
NM_003234:3430-3460   UUUAUCAGUGACAGAGUUCACUAUAAA
NM_014585:197-237     AACUUCAGCUACAGUGUUAGCUAAGUU
NM_003234:3884-3912   AUUAUCGGGAGCAGUGUCUUCCAUAAU
NM_003234:3481-3509   AUUAUCGGAAGCAGUGCCUUCCAUAAU
NM_000032:13-36       GU--UCGUCCUCAGUGCAGGGCA--AC
SS_cons               (((((.(((((......))))))))))
```

Reference: | **Example:**

Molecular control of vertebrate iron metabolism:
mRNA-basedregulatory circuits operated by iron,
nitric oxide, and oxidativestress.
Hentze MW, Kuhn LC;
Proc Natl Acad Sci U S A 1996;93:8175-8182.

Submitted by: | **Example:** Jason Wang

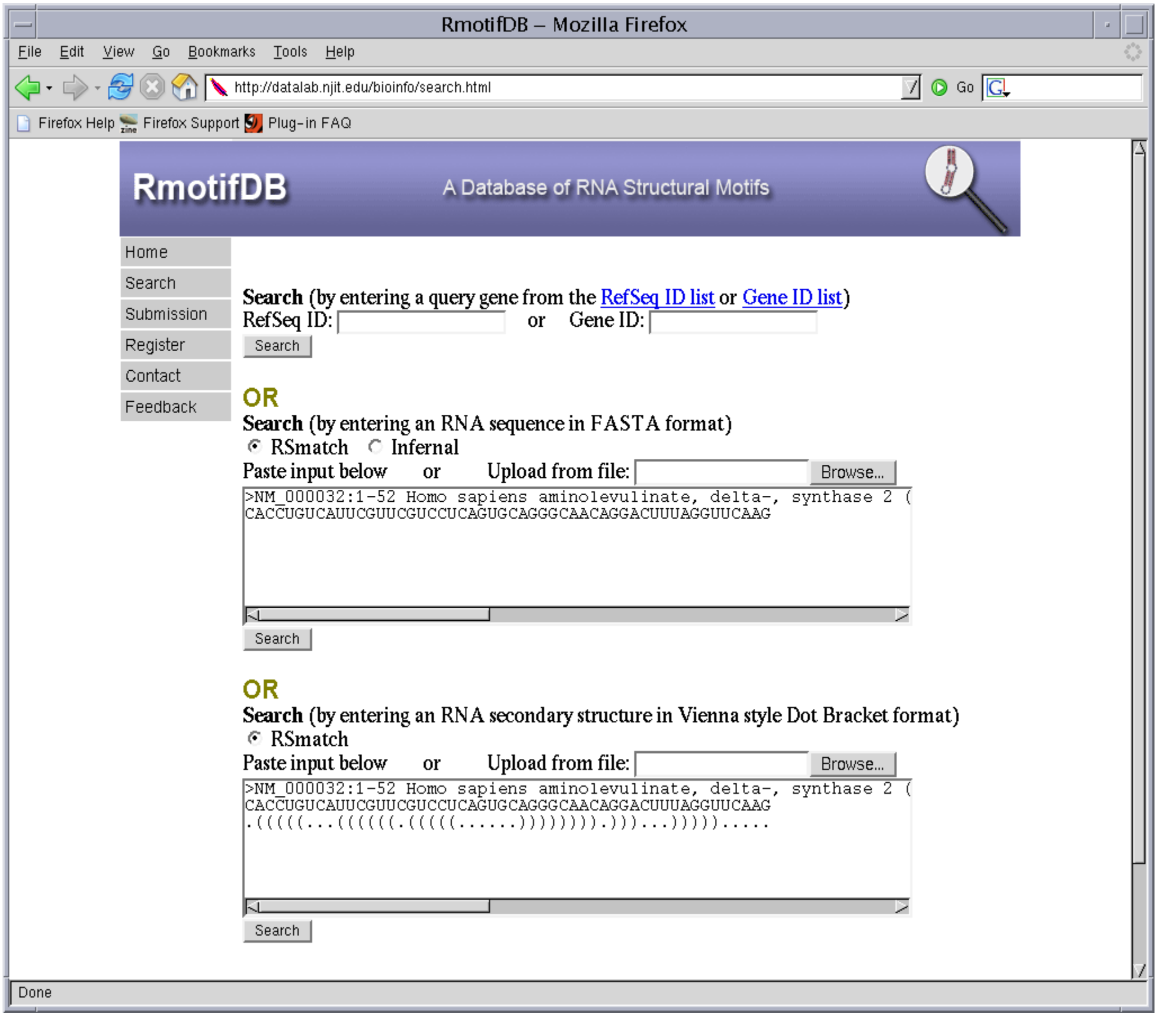
Email: | **Example:** wangj@njit.edu

Comment: | **Example:** Please enter your comment here. We appreciate your submission.

Figure 3.4 The interface of RmotifDB 2.0 where scientists can submit RNA structural motifs.

3.4 The RmotifDB 2.0 System

RmotifDB 2.0 is designed for storing the RNA structural motifs found in the UTRs of eukaryotic mRNAs. It is a web-based system that supports the retrieval and access of RNA structural motifs from its database. The system allows the user to search RNA structural motifs in an effective and user-friendly way. RmotifDB 2.0 is accessible at http://datalab.njit.edu/bioinfo/UTRdb/. It was implemented using Perl-CGI, Java, C and Oracle.


RmotifDB – Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://datalab.njit.edu/bioinfo/search.html
Go
Firefox Help
Firefox Support
Plug-in FAQ
RmotifDB
A Database of RNA Structural Motifs
Home
Search
Submission
Register
Contact
Feedback
Search
Browse...
Search
Browse...
Search
Done

Search (by entering a query gene from the RefSeq ID list or Gene ID list)
RefSeq ID: or Gene ID:

OR

Search (by entering an RNA sequence in FASTA format)
● RSmatch ○ Infernal
Paste input below or Upload from file:

```
>NM_000032:1-52 Homo sapiens aminolevulinate, delta-, synthase 2 (
CACCUGUCAUUCGUUCGUCCUCAGUGCAGGGCAACAGGACUUUAGGUUCAAG
```

OR

Search (by entering an RNA secondary structure in Vienna style Dot Bracket format)
● RSmatch
Paste input below or Upload from file:

```
>NM_000032:1-52 Homo sapiens aminolevulinate, delta-, synthase 2 (
CACCUGUCAUUCGUUCGUCCUCAGUGCAGGGCAACAGGACUUUAGGUUCAAG
.(((((...((((((.(((((......)))))))).)))...))))).....
```

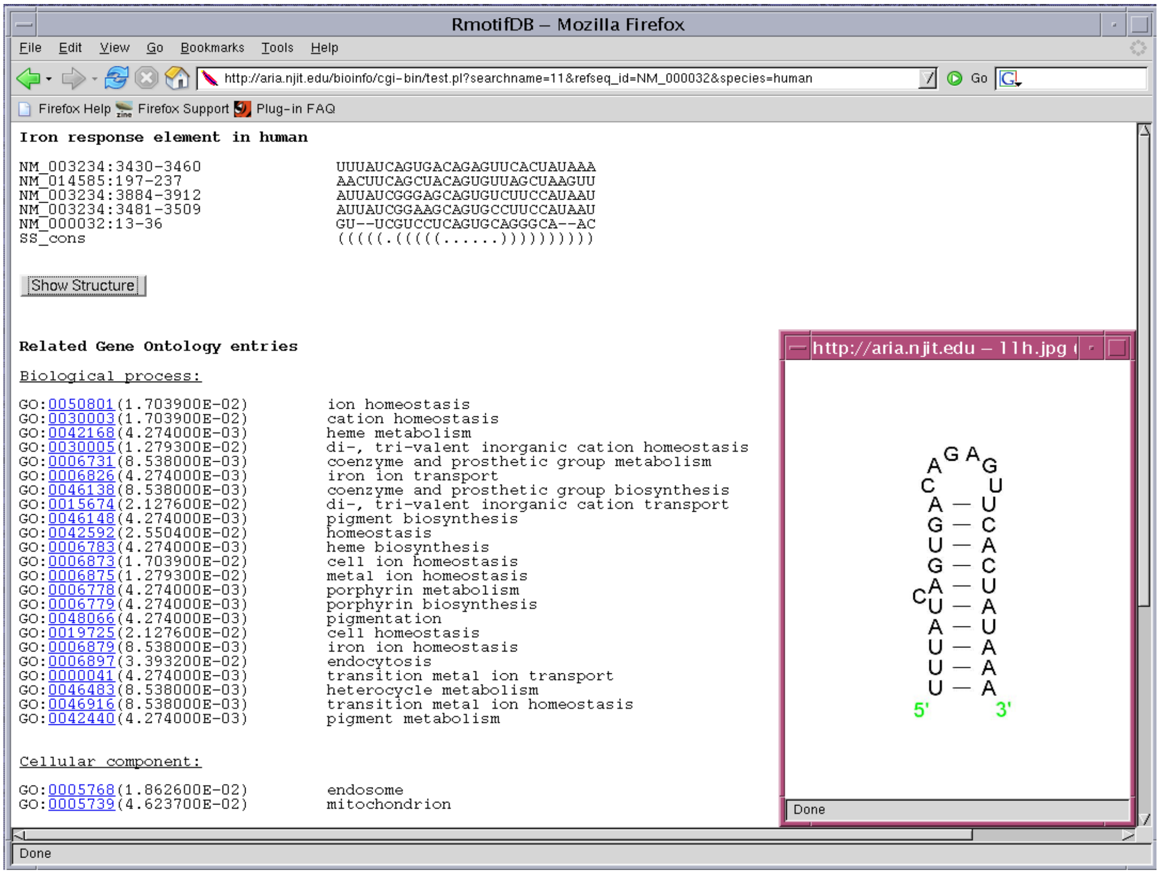
Figure 3.5 The search interface of RmotifDB 2.0 system.

The RNA structural motifs stored in RmotifDB 2.0 come from three sources. The primary source consists of manually collected motifs from the biomedical literature. Scientists who use this database can also submit motifs to RmotifDB 2.0. The interface scientists use to submit RNA structural motifs is shown in Figure 3.4. Lastly, motifs are obtained from those RNA structures discovered using a wide variety of motif mining methods (such as the method described in Section 3.3).

Figure 3.5 shows the search interface for RmotifDB 2.0. The system provides two search options: query-by-sequence (QBS) and query-by-structure (QBR). In QBS, the user enters an RNA sequence in the standard FASTA format [6] and the system matches this query sequence with motifs in the database using either RSmatch [9] or Infernal [13]. Since RSmatch accepts only RNA secondary structures as input data, the system needs to invoke Vienna RNA v1.4 [1] in order to fold the query sequence into a structure before a match is made. With QBR, the user enters an RNA secondary structure represented by the Vienna dot-bracket format [1] and the system matches this query structure with motifs in the database using RSmatch. The result is a ranked list of motifs that are approximately contained in the query sequence or the query structure. In addition, the user can search RmotifDB 2.0 by choosing a Gene ID or RefSeq ID from a pre-defined list of Gene IDs and RefSeq IDs provided by the RmotifDB 2.0 system where the Gene IDs and RefSeq IDs are obtained from http://www.ncbi.nlm.nih.gov/RefSeq/. This pre-defined list contains the IDs of the genes (mRNA sequences) used by several motif mining methods to discover the structural motifs stored in RmotifDB 2.0. The result of this search is a list of structural motifs containing the query gene ID (Gene ID or RefSeq ID).

3.5 Integrating RmotifDB 2.0 with Gene Ontology

While browsing the search results returned by RmotifDB 2.0, the user can click a motif to access pertinent detailed information. Figure 3.6 shows the result of displaying a motif and its related information. Here the motif is an iron response element (IRE) in humans shown in the Stockholm format [13]. This format is a multiple sequence alignment output with structural annotation in the Vienna dot-bracket format [1]. The motif is depicted in the bottom right-hand corner of the window. Also displayed is the Gene Ontology (GO) information concerning the motif, and relevant articles in PubMed (not shown in the screenshot) that publish said motif.


RmotifDB – Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://aria.njit.edu/bioinfo/cgi-bin/test.pl?searchname=11&refseq_id=NM_000032&species=human
Go
Firefox Help
Firefox Support
Plug-in FAQ
Show Structure
http://aria.njit.edu – 11h.jpg
A G A G
C U
A U
G C
U A
G C
A U
C
U A
A U
U A
U A
U A
5'
3'
Done
Done

Iron response element in human

```
NM_003234:3430-3460          UUUAUCAGUGACAGAGUUCACUAUAAA
NM_014585:197-237            AACUUCAGCUACAGUGUUAGCUAAGUU
NM_003234:3884-3912          AUUAUCGGGAGCAGUGUCUUCCAUAAU
NM_003234:3481-3509          AUUAUCGGAAGCAGUGCCUUCCAUAAU
NM_000032:13-36              GU--UCGUCCUCAGUGCAGGGCA--AC
SS_cons                      (((((.(((((......))))))))))
```

Related Gene Ontology entries

Biological process:

```
GO:0050801(1.703900E-02)        ion homeostasis
GO:0030003(1.703900E-02)        cation homeostasis
GO:0042168(4.274000E-03)        heme metabolism
GO:0030005(1.279300E-02)        di-, tri-valent inorganic cation homeostasis
GO:0006731(8.538000E-03)        coenzyme and prosthetic group metabolism
GO:0006826(4.274000E-03)        iron ion transport
GO:0046138(8.538000E-03)        coenzyme and prosthetic group biosynthesis
GO:0015674(2.127600E-02)        di-, tri-valent inorganic cation transport
GO:0046148(4.274000E-03)        pigment biosynthesis
GO:0042592(2.550400E-02)        homeostasis
GO:0006783(4.274000E-03)        heme biosynthesis
GO:0006873(1.703900E-02)        cell ion homeostasis
GO:0006875(1.279300E-02)        metal ion homeostasis
GO:0006778(4.274000E-03)        porphyrin metabolism
GO:0006779(4.274000E-03)        porphyrin biosynthesis
GO:0048066(4.274000E-03)        pigmentation
GO:0019725(2.127600E-02)        cell homeostasis
GO:0006879(8.538000E-03)        iron ion homeostasis
GO:0006897(3.393200E-02)        endocytosis
GO:0000041(4.274000E-03)        transition metal ion transport
GO:0046483(8.538000E-03)        heterocycle metabolism
GO:0046916(8.538000E-03)        transition metal ion homeostasis
GO:0042440(4.274000E-03)        pigment metabolism
```

Cellular component:

```
GO:0005768(1.862600E-02)        endosome
GO:0005739(4.623700E-02)        mitochondrion
```

Figure 3.6 The output showing a structural motif stored in RmotifDB 2.0 and related information. The *t*-value inside the parentheses next to each GO entry indicates the significance of the association between the motif and the GO entry. The smaller the *t*-value, the more significant the association.

In general, a motif contains multiple genes (mRNA sequences) with similar functions. The GO entries and their URLs that are highly associated with the motif are collected and stored in RmotifDB 2.0. The GO entries belonging to three categories (molecular function, biological process and cell component) are obtained from the Gene Ontology Consortium (http://www.geneontology.org). The mapping information between the GO entries and the genes is obtained from the LocusLink database [35]. A hyper-geometric test [36] is used to measure the significance of the association between the motif and each of the GO entries. The significance is shown as the parenthesized *t*-value next to each GO entry in Figure 3.6. The hyper-geometric test is appropriate as considering a finite population sampling scheme with the entire population divided into two groups: those associated with a particular GO entry and those associated with the other GO entries.

Generally speaking, the hyper-geometric test has four parameters (which shall be related to the problem in a moment):

- *m*, the number of white balls in an urn,
- *n*, the number of black balls in the urn,
- *k*, the number of balls drawn from the urn,
- *x*, the number of white balls drawn from the urn.

The probability that *x* out of *k* balls drawn from the urn are white (from an urn containing *m*+*n* balls) is:

$$f(x,m,n,k) = \frac{\binom{m}{x}\binom{n}{k-x}}{\binom{m+n}{k}} \tag{3.1}$$

where $x \leq \min(m, k)$.

For each RNA structural motif *M* containing multiple genes, all GO entries are examined to evaluate their associations with *M*. Through the mapping information between *M* and a GO entry *G*, in a GO category *C*, it is able to calculate four values:

- N_1, the number of genes associated with any GO entry in *C*,
- N_2, the number of genes associated with *G* in *C*,
- N_3, the number of genes in *M* associated with any GO entry in *C*,
- N_4, the number of genes in *M* associated with *G* in *C*,

where $N_1 \geq N_2$ and $N_3 \geq N_4$.

The *t*-value of the GO entry *G* is calculated as:

$$t(G) = f(N_4, N_2, N_1\text{-}N_2, N_3) \qquad (3.2)$$

where the function *f* is defined in Equation (3.1). In general, the smaller the value of *t*(*G*), the more significant the association between *G* and *M*. RmotifDB 2.0 displays *G* together with its *t*-value, if *t*(*G*) is smaller than a user-adjustable parameter value (0.05 in the present case).

3.6 Conclusions

In this chapter, an advanced RNA structural motif database called RmotifDB 2.0 was presented and some of its features were described, as well as techniques used for integrating RmotifDB 2.0 with Gene Ontology. A motif mining method capable of discovering structural motifs in eukaryotic mRNAs was developed. Data mining [18,20] and data integration [37-46] have emerged as important fields in bioinformatics at the interface of information technology and molecular biology. The system presented is part of a long-term project [14,47] that aims to build a cyberinfrastructure for RNA data mining and data integration. This cyberinfrastructure complements existing RNA motif databases such as Rfam and UTRdb [4,48] which lack structure-based search functions. It contributes to this field in general and to RNA informatics in particular.

CHAPTER 4

THE VISUALIZATION OF RSMATCH

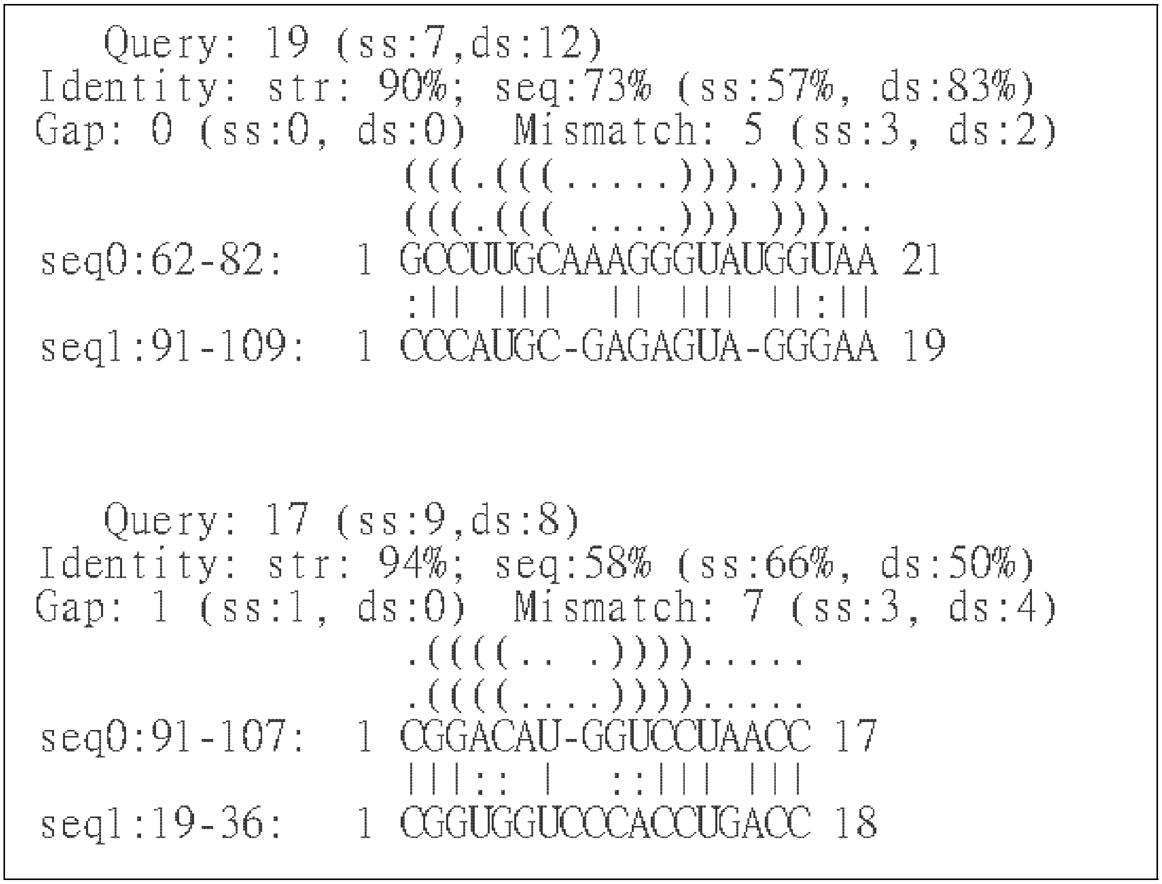
4.1 Preface

This chapter describes RSview, a tool for graphically displaying the alignment results produced by RSmatch [9]. Figure 4.1 illustrates a sample output of pairwise sequence alignments generated by RSmatch. Its output is presented in plain text format. Since the plain text format of RSmatch output may be inconvenient for researchers interested in RNA structural motifs, it is important to have a software tool which can present the output in a visually effective graphical manner.

4.2 RSview

A visualization tool called RSview was developed, which is used with RSmatch that re-displays RSmatch's plain text output of alignment results. Given two RNA molecules, RSview displays the RSmatch's output in a colored, graphical manner by integrating RNAView [49] with RSmatch. The programming languages used to implement RSview are C, Java and Perl.

The function of the RNAView program is to generate 2-dimensional (2D) figures of DNA/RNA secondary structures including tertiary interactions. RNAView is able to identify and classify the types of base pairs formed in nucleic acid structures. The RNAView program accepts RNA structures with 3-dimensional (3D) coordinate data in PDB, mmCIF or RNAML format and produces 2D diagrams of secondary and tertiary RNA structures in Postscript, VRML or RNAML formats. Figures 4.2 and 4.3 show example diagrams produced by RNAView with PDB ID's 1GID Chain A (P4-P6 RNA RIBOZYME DOMAIN) and 1C2X Chain C (5S RIBOSOMAL RNA), respectively.



```
   Query: 19 (ss:7,ds:12)
Identity: str: 90%; seq:73% (ss:57%, ds:83%)
Gap: 0 (ss:0, ds:0)  Mismatch: 5 (ss:3, ds:2)
                (((.(((.....))).)))..
                (((.((( ....))) )))..
seq0:62-82:   1 GCCUUGCAAAGGGUAUGGUAA 21
                :|| |||  || ||| ||:||
seq1:91-109:  1 CCCAUGC-GAGAGUA-GGGAA 19



   Query: 17 (ss:9,ds:8)
Identity: str: 94%; seq:58% (ss:66%, ds:50%)
Gap: 1 (ss:1, ds:0)  Mismatch: 7 (ss:3, ds:4)
                .((((.. .)))).....
                .((((....)))).....
seq0:91-107:  1 CGGACAU-GGUCCUAACC 17
                |||:: |  ::||| |||
seq1:19-36:   1 CGGUGGUCCCACCUGACC 18
```

Figure 4.1 One sample output of pairwise sequence alignment of RSmatch.

Like RNAView, RSview accepts a pair of RNA structures with 3D coordinate data in PDB format and generates a pair of 2D diagrams in Postscript format, as illustrated in Figures 4.2 and 4.3, together with RNA structures with 3D coordinate data in RNAML format for the input RNA molecules. The sequences of the input RNA molecules are extracted from the RNAML files and folded into the secondary structures using the Vienna RNA package [1]. These secondary structures are then aligned using RSmatch. Finally RSview combines the two simplified 2D Postscript format diagrams for the RNA molecules with the alignment results obtained from RSmatch.

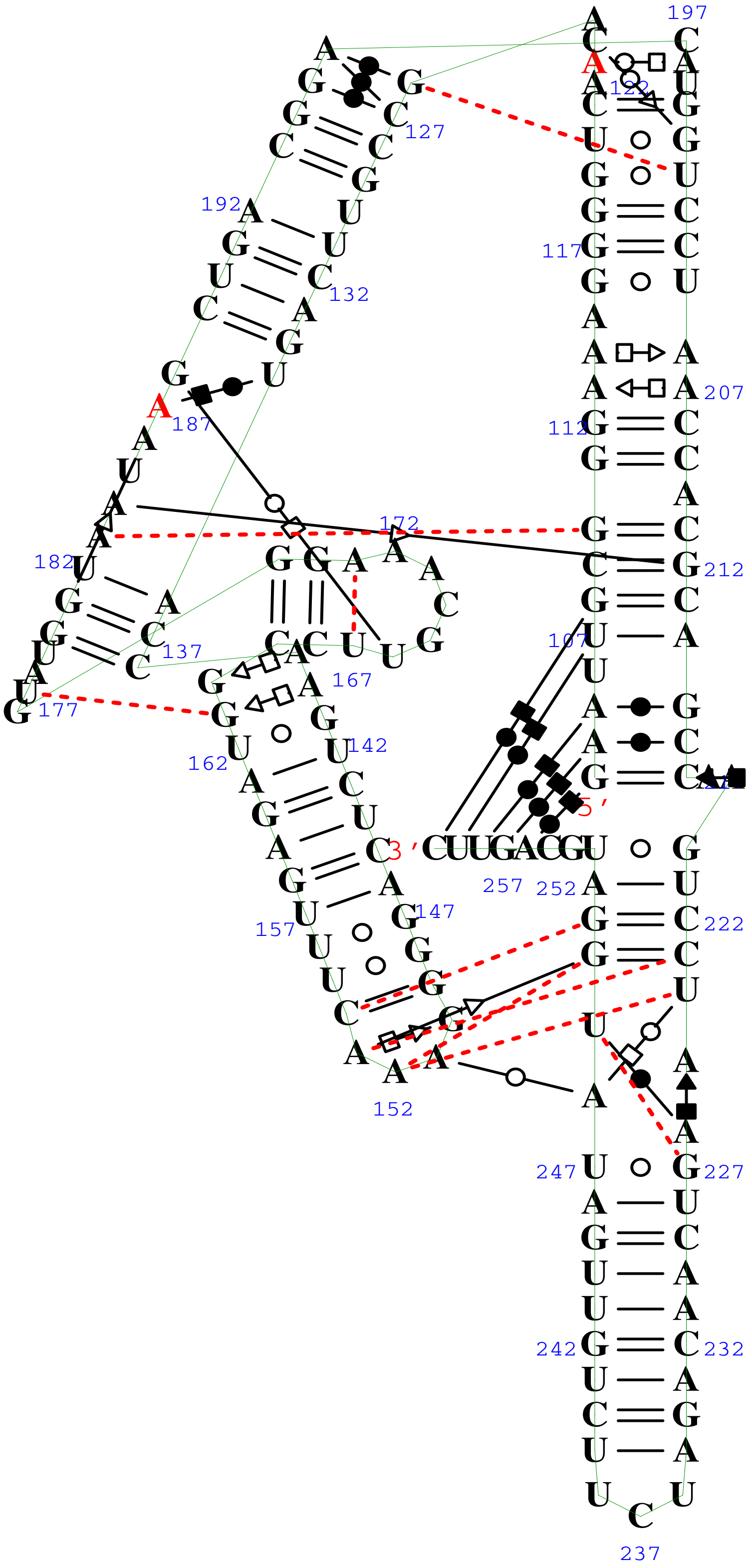


Figure 4.2 The diagram of 1GID Chain A (P4-P6 RNA RIBOZYME DOMAIN) produced by RNAView [49].

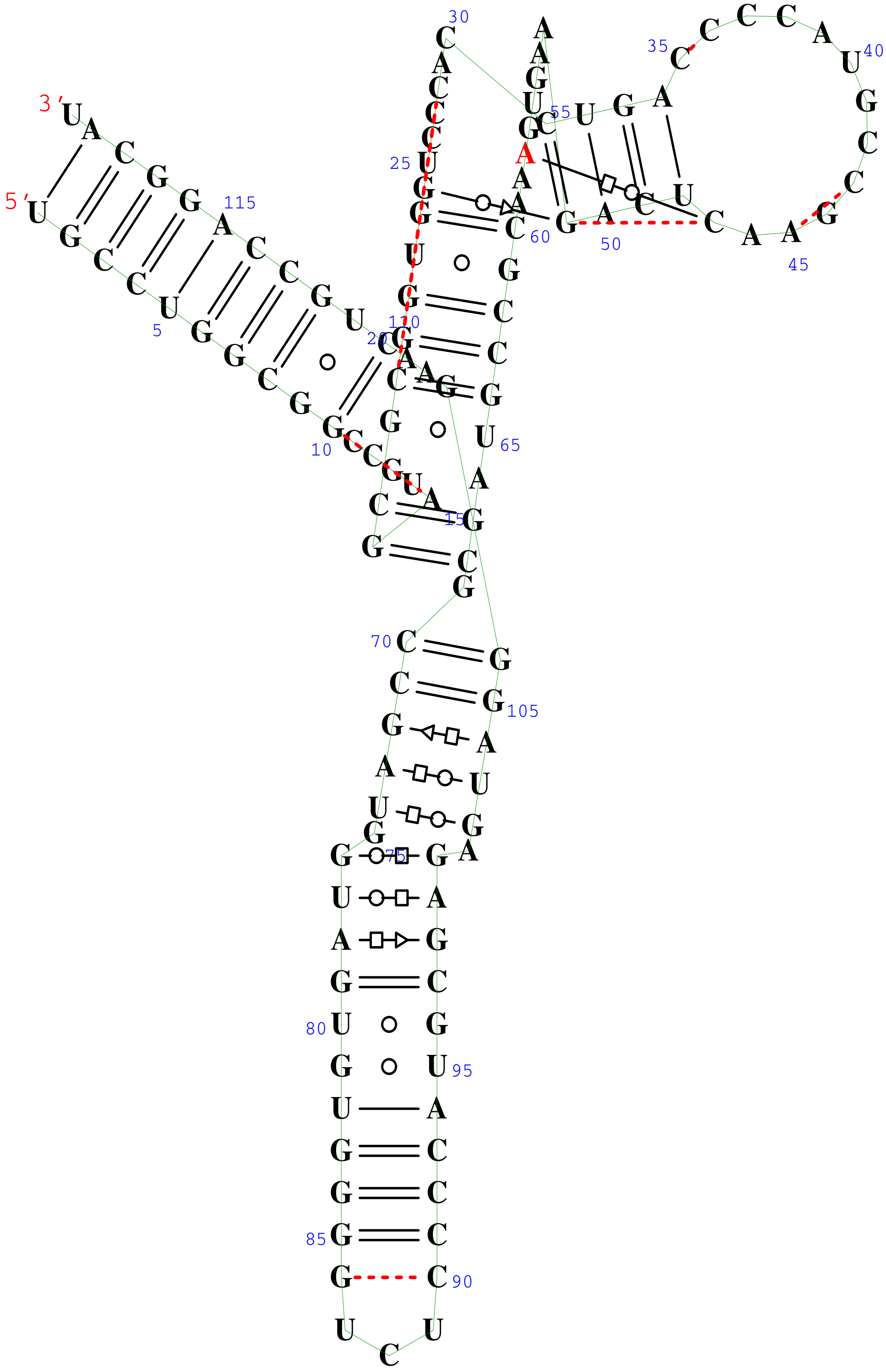


Figure 4.3 The diagram of 1C2X Chain C (5S RIBOSOMAL RNA) produced by RNAView [49].

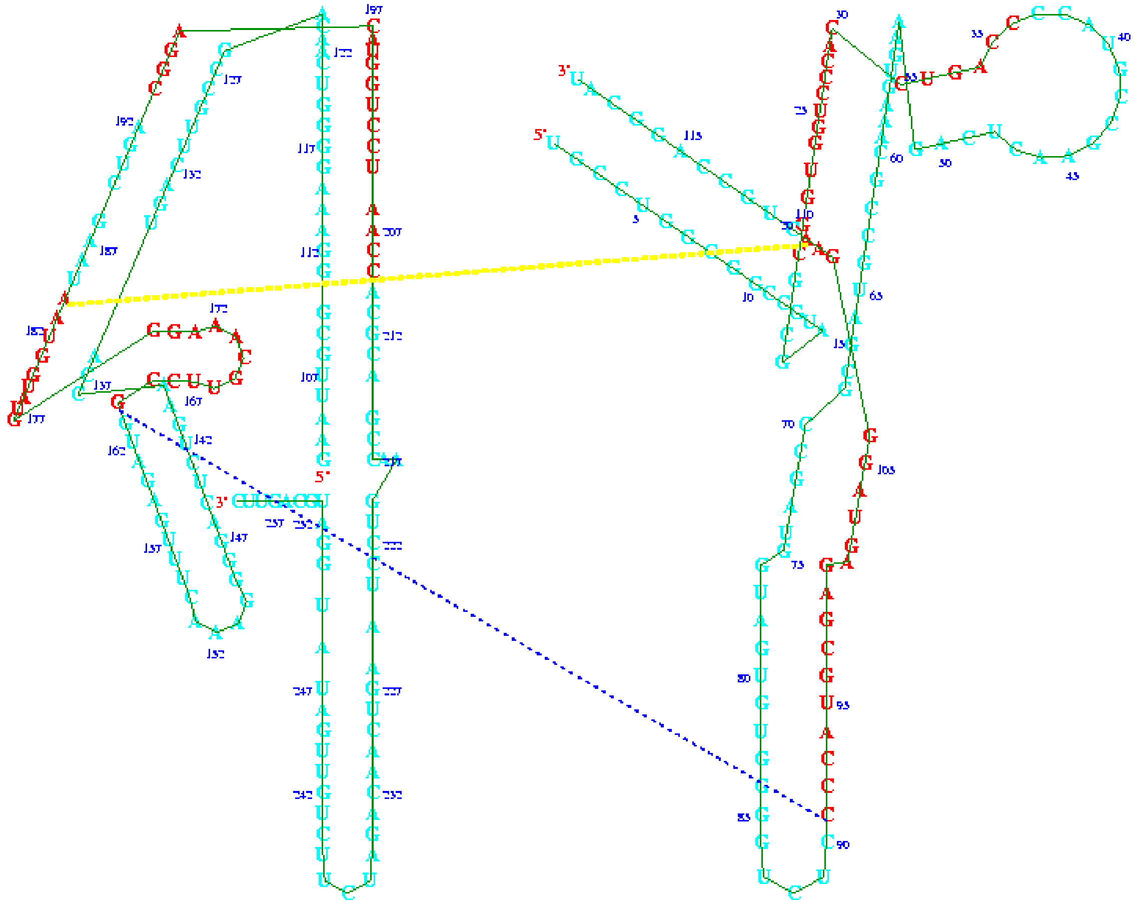


Figure 4.4 The output diagram of RSview.

Figure 4.4 shows the output of RSview for the two molecules in Figures 4.2 and 4.3. In Figure 4.4, the nucleotides in cyanine color are the unmatched region and the nucleotides in red are the matched (aligned) region. The blue (starting) line and yellow (ending) line indicate the best local match with the largest alignment score among all matched (aligned) regions. The web version of RSview with tutorial is available at http://datalab.njit.edu/biodata/rna/RSview/.

CHAPTER 5

PREDICTING COAXIAL HELICAL STACKING IN RNA JUNCTIONS

5.1 Preface

In previous chapters, the topics mainly focused on the RNA secondary structures and their motifs in 2D. However, since the advanced improvement of the crystallography on the RNA molecules in recent years, the study, analysis and prediction on the RNA tertiary structures has become extensive [50-52], which had not been possible over the past several decades. Therefore, beginning with this chapter, the topic will focus on the study of RNA tertiary structures and their three-dimensional (3D) motifs.

It is well-known that the RNA junction is one of the essential structural components in RNA molecules. The RNA junction is formed by at least three helices in RNA tertiary structures. In order to explore the analysis and prediction of the RNA tertiary structure, it is important to study the structural configuration of the RNA junctions.

In this chapter, a data mining method is described to predict the configuration of the coaxial helical stacks and families (topologies) in RNA 3-way to 10-way junctions at the secondary structure level. This method adopts the random forests classifier which is trained by solved RNA tertiary structures. In Section 5.2, the background knowledge of the coaxial stacking and the family (topology) on RNA junctions is introduced. The details of the materials and methods used for prediction are described in Section 5.3. To ensure the accuracy and performance of the proposed method, the experiments and the performance evaluations, including the comparison with other works, are reported in Section 5.4. Furthermore, the features, which are extracted from the junctions and used for prediction, are analyzed for future improvement. The analysis for the features is discussed in Section 5.5.

5.2 Background

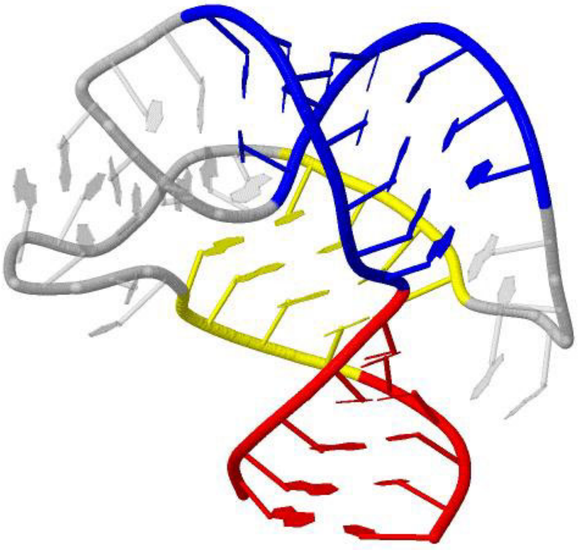
An RNA molecule is composed of many different components such as helices, hairpin loops, bulge/internal loops, pseudoknots and junctions. An RNA junction, also known as a multi-branch loop, can be defined as the enclosed area composed of more than two helical segments [53,54]. This structural component of RNA can be found in numerous RNA molecules, and is used in a wide range of functional roles such as the self-cleaving catalytic domain of the hammerhead ribozyme. Due to the fact that junctions play a role as major architectural components in RNA, understanding the structural properties of junctions is necessary.

A common tertiary motif among junctions is the coaxial stacking of helices [50,55]. This motif is formed when two separate helical elements stack to form coaxial helices as a pseudo-continuous helix. Coaxial stacking motifs are seen in several large RNA structures, including tRNA, group II intron, and the large ribosomal subunits. Coaxial stacking provides thermodynamic stability to the molecule, and reduces the separation between loop regions in junctions. Both coaxial stacking and long-range interactions are essential for the correct tertiary structure formation of many RNAs as well as the formation of different junction topologies [50,56,57].

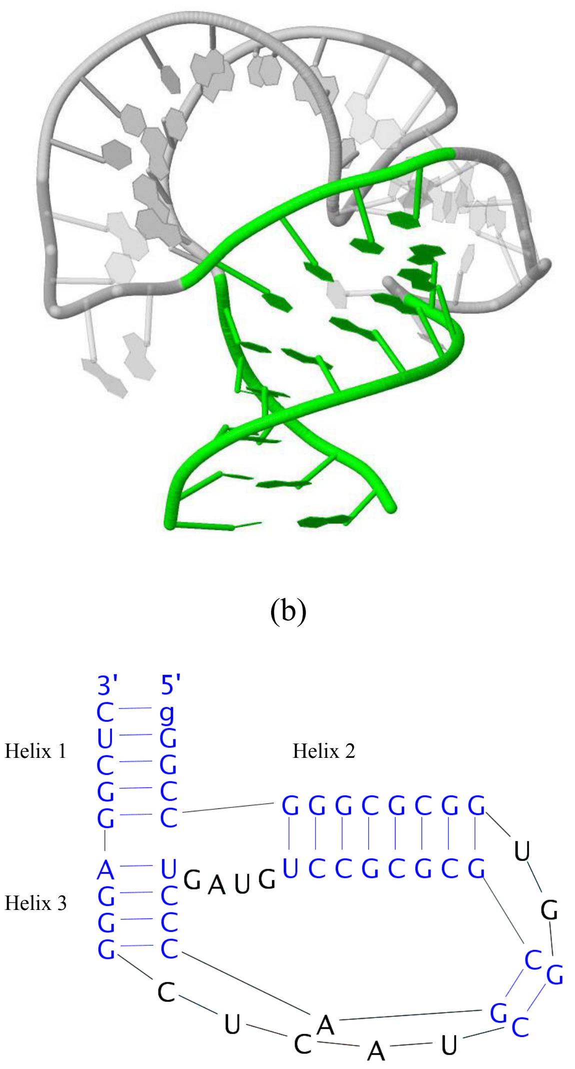
Analyses from solved crystal structures have shown that, according to their three-dimensional shape or topology, RNA junctions can be categorized into several families. Specifically, Lescoute and Westhof compiled and analyzed the topology of three-way junctions in folded RNAs, grouping these junctions into three families A, B, and

C [58]. In most of the structured three-way junctions, two of the helices stack coaxially. Laing and Schlick analyzed RNA four-way junctions and grouped them into nine families such as H, cH, cL, cK, π, cW, ψ, cX, and X, according to coaxial stacking interactions and helical conformation signatures [56].

One example of an RNA molecule (PDB ID: 1E8O) with a three-way junction is presented in Figure 5.1. The three-dimensional view is rendered by Jmol (http://www.jmol.org/). The secondary structure view of the same molecule is rendered by S2S [59]. In this figure, each helix of the three-way junction is highlighted by different colors as well as the coaxial stacking. It is clearly shown where two separated helices stack and form the helical coaxial stacking as a pseudo-continuous helix. Another example of an RNA molecule (PDB ID: 3DIL) with a five-way junction is presented in Figure 5.2. There are two coaxial stacks formed in this five-way junction. One highlighted in light green is between Helix 1 and Helix 2. Another highlighted in orange is between Helix 4 and Helix 5.



(a)



(b)

(c)

Figure 5.1 (a) An RNA molecule (PDB ID: 1E8O) with a three-way junction is rendered by Jmol. Helix 1 is highlighted in red. Helix 2 is highlighted in blue. Helix 3 is highlighted in yellow. (b) A coaxial stacking is formed by Helix 1 and Helix 3. It is highlighted in light green. (c) The secondary structure view of this RNA molecule is rendered by S2S. Helix 1, Helix2 and Helix 3 are labeled. Helix 1 and Helix 3 are aligned, which represents the formation of the coaxial stacking.

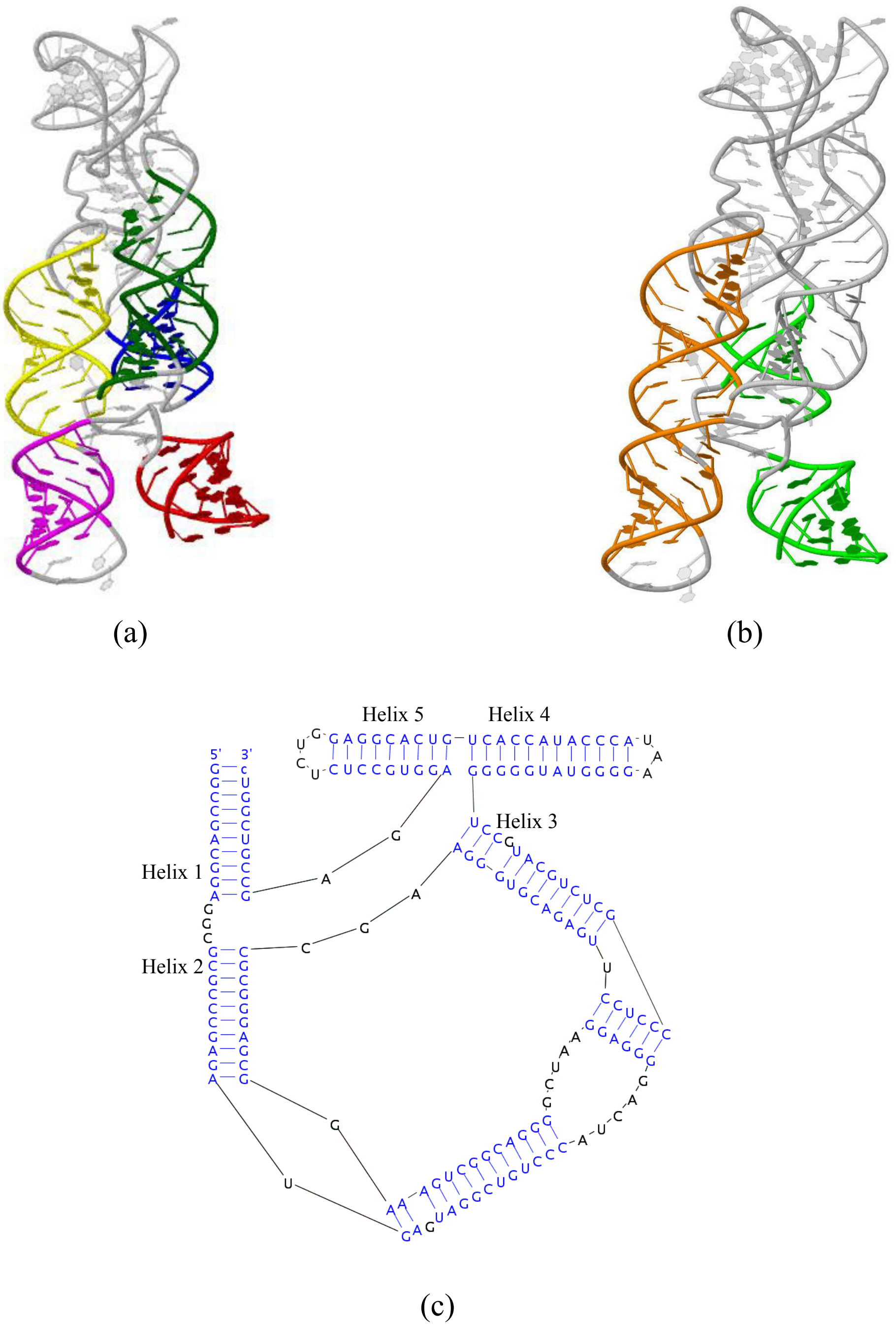


Figure 5.2 (a) An RNA molecule (PDB ID: 3DIL) with a five-way junction is rendered by Jmol. Helix 1 is highlighted in red. Helix 2 is highlighted in blue. Helix 3 is highlighted in dark green. Helix 4 is highlighted in yellow. Helix 5 is highlighted in magenta. (b) One coaxial stacking highlighted in light green is formed by Helix 1 and Helix 2. Another coaxial stacking highlighted in orange is formed by Helix 4 and Helix 5. (c) The secondary structure view of this RNA molecule is rendered by S2S. Helix 1, Helix2, Helix 3, Helix 4 and Helix 5 are labeled. Helix 1 and Helix 2 are aligned as well as Helix 4 and Helix 5, which represent the formation of the coaxial stacks.

5.3 Materials and Methods

Dataset of RNA junctions, feature extraction and random forests algorithm are described and discussed in this section.

5.3.1 Dataset of RNA Junctions

The dataset of RNA junctions used in this study is the updated dataset from Laing's previous works [56,57]. The dataset is collected from the 3D RNA structures of the RCSB Protein Data Bank [12] as of November 2010. A total of 216 RNA junctions were collected. The information of coaxial stacking and junction family (topology) are manually entered into the dataset. Figure 5.3 shows the number of junctions for each junction order. In this dataset, only the Watson-Crick (AU, GC) and Wobble (GU) base pairs are considered and a helix is defined as at least two consecutive base pairs. On each helix of a junction, the two consecutive base pairs closing the junction and all single bases between the helices are collected into the dataset, which defines the scope of a junction as shown in Figure 5.4 for example.

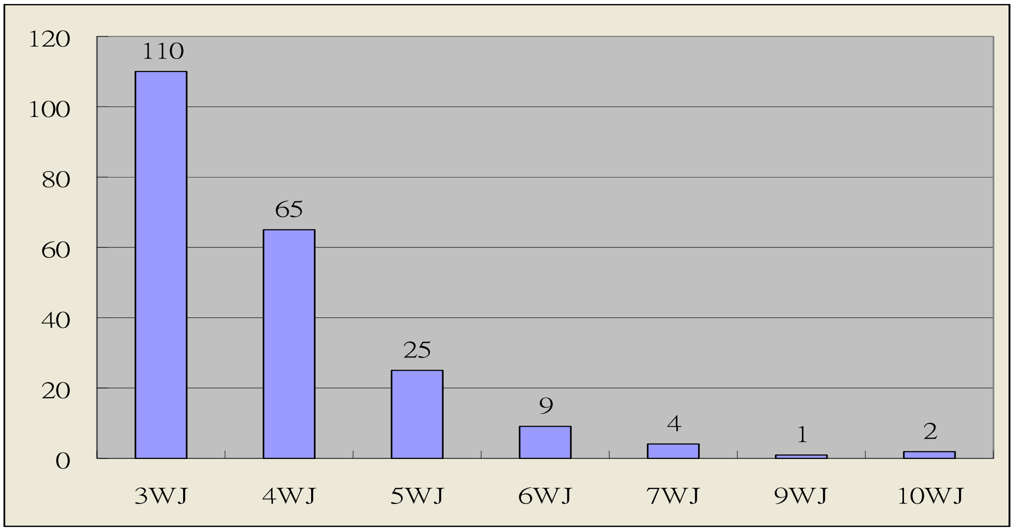


Figure 5.3 The number of junctions for each junction order.

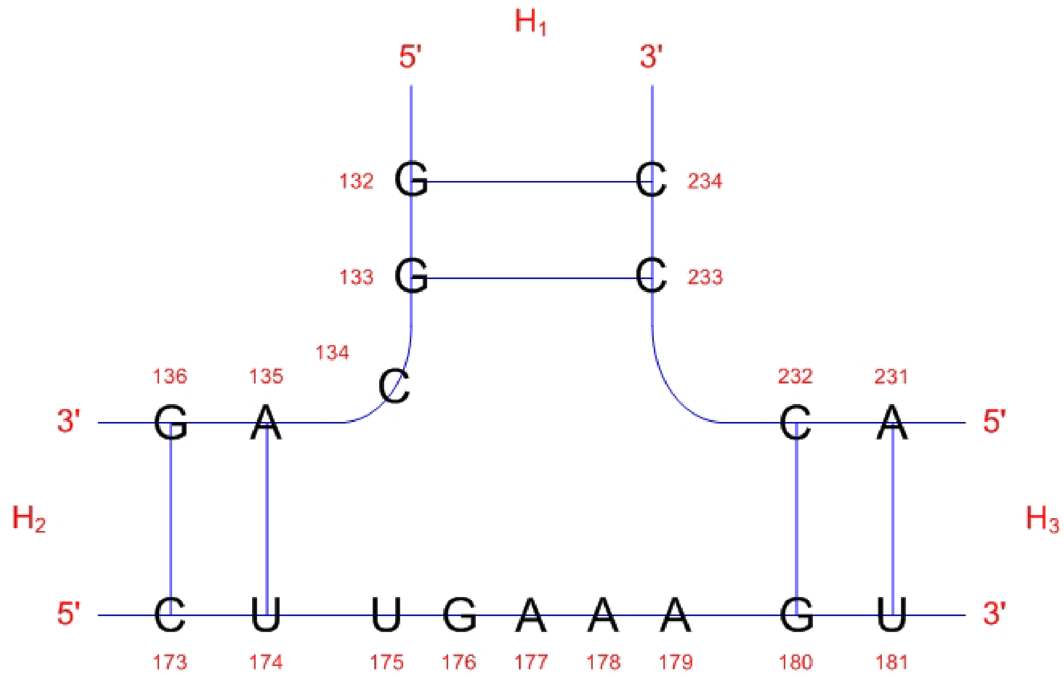


Figure 5.4 The scope of a three-way junction. On each helix of a junction, the two consecutive base pairs closing the junction are collected, as well as all single bases between helices.

The dataset contains tables for each junction order. For example, the attributes of the table for the three-way junction are described as follows. The attribute **PDB** represents the PDB (Protein Data Bank) ID of the RNA molecule in which the junction is collected. The **RNA Type** attribute is the type of RNA molecule in which the junction is collected. The family (topology) type of each three-way junction is recorded under the **Family** attribute. The family type of each three-way junction is either A, B or C. The **Coaxial** attribute is the coaxial stacking configuration of each junction. For three-way junctions, there are four different types of coaxial stacking including H_1H_2, H_2H_3, H_1H_3 and None. H_1, H_2 and H_3 represent the first, second and third helix of a three-way junction respectively. H_1H_2 represents a coaxial stacking formed by the first helix (Helix 1) and the second helix (Helix 2) in a junction. H_2H_3 represents a coaxial stacking formed by the second helix (Helix 2) and the third helix (Helix 3) in a junction. H_1H_3 represents a coaxial stacking formed by the first helix (Helix 1) and the third helix (Helix 3) in a junction. Therefore, “None” represents no coaxial stacking formed in a junction.

A three-way junction is completely described by attributes representing three RNA subsequences. The position numbers and the nucleotides (A, U, C, G) are given for each RNA subsequence. The attributes **StrSeq1**, **StrSeq2** and **StrSeq3** represent the first, second and third RNA subsequences. The starting and ending position numbers of the first subsequence are named **S1ID5** and **S1ID3** which indicate the 5’ and 3’ ends of the first subsequence. The position numbers of the second subsequence are attributes **S2ID5** and **S2ID3**. The position numbers of the third subsequence are attributes **S3ID5** and **S3ID3**. By taking the three-way junction shown in Figure 5.4 as an example, **S1ID5** is 132, **S1ID3** is 136, **S2ID5** is 173, **S2ID3** is 181, **S3ID5** is 231 and **S3ID3** is 234. Similarly, the first subsequence (**StrSeq1**) is GGCAG, the second subsequence (**StrSeq2**) is CUUGAAAGU and the third subsequence (**StrSeq3**) is ACCC.

Single/unpaired bases between helices in the junction and the number of these bases are shown as attributes $\mathbf{J_{12}}$, $\mathbf{J_{23}}$ and $\mathbf{J_{31}}$. For example, in Figure 5.4, $\mathbf{J_{12}}$ represents the unpaired bases between the first and second helices, which is C. Therefore, $\mathbf{J_{23}}$ is UGAAA and $\mathbf{J_{31}}$ is blank. Therefore, the lengths of attributes $\mathbf{J_{12}}$, $\mathbf{J_{23}}$ and $\mathbf{J_{31}}$ are 1, 5 and 0 respectively.

5.3.2 Feature Extraction

The dataset of junctions from solved RNA molecules are used for training by the random forests algorithm. A trained random forests classifier will be used to predict the helical coaxial stacking and junction family types. Since there are training and testing phases in the random forests algorithm, it is necessary to extract features from the dataset of junctions. The information, including the loops length between helices, sequence content and thermodynamic free-energy associated with the base pairs on helices and their common loop region, is extracted from the secondary structure level as features.

Table 5.1 lists the 15 features for three-way junctions. Because coaxial stacking is favorable to smaller loop region length, all loop region lengths ($|J_{12}|$, $|J_{23}|$, $|J_{13}|$), their ascending order ($\mathrm{Min}(|J_{12}|,|J_{23}|,|J_{13}|)$, $\mathrm{Med}(|J_{12}|,|J_{23}|,|J_{13}|)$, $\mathrm{Max}(|J_{12}|,|J_{23}|,|J_{13}|)$) and the smaller length of two neighboring loop regions ($\mathrm{Min}(|J_{23}|,|J_{13}|)$, $\mathrm{Min}(|J_{12}|,|J_{13}|)$, $\mathrm{Min}(|J_{12}|,|J_{23}|)$) are considered as features. Furthermore, the maximum number of consecutive adenines in the loop region ($A(J_{12})$, $A(J_{23})$ $A(J_{13})$) is also considered since it has been reported that adenines in loop regions often form tertiary motifs named A-minor [60] in specific junction topologies [56-58].

To improve the prediction accuracy of coaxial stacking in junctions, thermodynamic free-energy associated with terminal base pairs on two neighboring helices and their common loop region is considered ($\Delta G(H_1,H_2)$, $\Delta G(H_2,H_3)$, $\Delta G(H_1,H_3)$). When the length of the loop region is 0 or 1, the thermodynamic free-energy values are taken from the tables of the program RNAstructure [61]. When the length of the loop region is 0, the free-energy value is taken from the table of coaxial stacking for two helices with no intervening unpaired nucleotide. When the length of the loop region is 1, the free-energy value is calculated from the table of coaxial stacking with one intervening mismatch and plus 2.1 kcal/mol for the terminal mismatch free-energy, as suggested by Tyagi and Mathews [62]. As a terminal mismatch in J_i can potentially form a non-canonical base-pair with a nucleotide in J_{i-1} or J_{i+1}, the minimum free-energy value is considered for these two possibilities.

Table 5.1 Features Used for Predicting Helical Coaxial Stacking and Topology of Three-way Junctions

Feature	Description
$\lvert J_{12}\rvert$	Number of nucleotides in the loop region between helix H_1 and helix H_2
$\lvert J_{23}\rvert$	Number of nucleotides in the loop region between helix H_2 and helix H_3
$\lvert J_{13}\rvert$	Number of nucleotides in the loop region between helix H_1 and helix H_3
$\mathrm{Min}(\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{13}\rvert)$	The minimum value of $\lvert J_{12}\rvert$, $\lvert J_{23}\rvert$ and $\lvert J_{13}\rvert$
$\mathrm{Med}(\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{13}\rvert)$	The median value of $\lvert J_{12}\rvert$, $\lvert J_{23}\rvert$ and $\lvert J_{13}\rvert$
$\mathrm{Max}(\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{13}\rvert)$	The maximum value of $\lvert J_{12}\rvert$, $\lvert J_{23}\rvert$ and $\lvert J_{13}\rvert$
$\mathrm{Min}(\lvert J_{23}\rvert,\lvert J_{13}\rvert)$	Minimum value of $\lvert J_{23}\rvert$ and $\lvert J_{13}\rvert$
$\mathrm{Min}(\lvert J_{12}\rvert,\lvert J_{13}\rvert)$	Minimum value of $\lvert J_{12}\rvert$ and $\lvert J_{13}\rvert$
$\mathrm{Min}(\lvert J_{12}\rvert,\lvert J_{23}\rvert)$	Minimum value of $\lvert J_{12}\rvert$ and $\lvert J_{23}\rvert$
$A(J_{12})$	Maximum number of consecutive adenines in the loop region between helix H_1 and helix H_2
$A(J_{23})$	Maximum number of consecutive adenines in the loop region between helix H_2 and helix H_3
$A(J_{13})$	Maximum number of consecutive adenines in the loop region between helix H_1 and helix H_3
$\Delta G(H_1,H_2)$	Thermodynamic free-energy associated with helix H_1, helix H_2 and the loop region between H_1 and H_2
$\Delta G(H_2,H_3)$	Thermodynamic free-energy associated with helix H_2, helix H_3 and the loop region between H_2 and H_3
$\Delta G(H_1,H_3)$	Thermodynamic free-energy associated with helix H_1, helix H_3 and the loop region between H_1 and H_3

Table 5.2 Features Used for Predicting Helical Coaxial Stacking on a Pair of Neighboring Helices H_i and H_{i+1} in Higher-order Junctions

Feature	Description
$\lvert J_{i(i+1)}\rvert$	Number of nucleotides in the loop region between helix H_i and helix H_{i+1}
$\lvert J_{(i-1)i}\rvert$	Number of nucleotides in the loop region between helix H_{i-1} and helix H_i
$\lvert J_{(i+1)(i+2)}\rvert$	Number of nucleotides in the loop region between helix H_{i+1} and helix H_{i+2}
$\mathrm{Min}(\lvert J_{(i-1)i}\rvert,\lvert J_{(i+1)(i+2)}\rvert)$	Minimum value of $\lvert J_{(i-1)i}\rvert$ and $\lvert J_{(i+1)(i+2)}\rvert$
$A(J_{i(i+1)})$	Maximum number of consecutive adenines in the loop region between helix H_i and helix H_{i+1}
$A(J_{(i-1)i})$	Maximum number of consecutive adenines in the loop region between helix H_{i-1} and helix H_i
$A(J_{(i+1)(i+2)})$	Maximum number of consecutive adenines in the loop region between helix H_{i+1} and helix H_{i+2}
$\Delta G(H_i,H_{i+1})$	Thermodynamic free-energy associated with helix H_i, helix H_{i+1} and the loop region between H_i and H_{i+1}
$\Delta G(H_{i-1},H_i)$	Thermodynamic free-energy associated with helix H_{i-1}, helix H_i and the loop region between H_{i-1} and H_i
$\Delta G(H_{i+1},H_{i+2})$	Thermodynamic free-energy associated with helix H_{i+1}, helix H_{i+2} and the loop region between H_{i+1} and H_{i+2}

As it is currently impossible to calculate the thermodynamic parameters by wet lab experiments for any loop region length greater than one, the thermodynamic free-energy values are estimated by a linear or a logarithmic equation [63,64] as follows. When the length of the loop region is between 2 and 6, the free-energy value is calculated as:

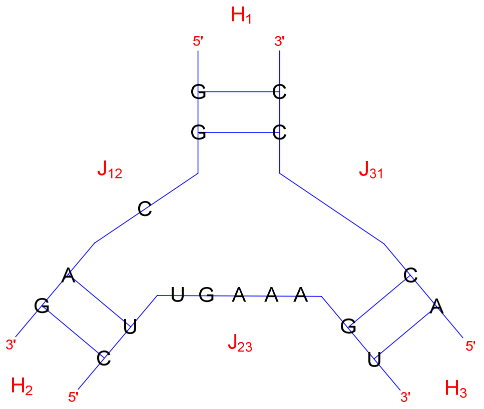
$$a + bL + ch \tag{5.1}$$

where a = 9.3, b = -0.3, c = -0.9, h = 2 and L is the length of the loop region. When the length of the loop region is greater than 6, the free-energy value is calculated as:

$$a + 6b + 1.1\ln(L/6) + ch \tag{5.2}$$

where a = 9.3, b = -0.3, c = -0.9, h = 2 and L is the length of loop region.

Figure 5.5 illustrates an example of a three-way junction and its 15 feature values. Similarly, the four-way junction is associated with 18 feature values. However, for five-way or higher-order junctions, the features are determined "locally". Since there is less data for higher-order junctions than for three-way and four-way junctions, a common feature set for all higher-order junctions is necessary.

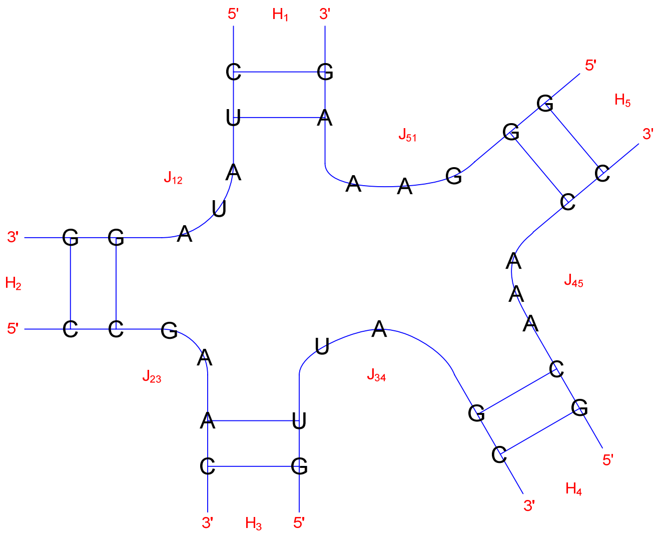


Feature Name	Feature Value
$\lvert J_{12}\rvert$	1
$\lvert J_{23}\rvert$	5
$\lvert J_{13}\rvert$	0
$\mathrm{Min}(\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{13}\rvert)$	0
$\mathrm{Med}(\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{13}\rvert)$	1
$\mathrm{Max}(\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{13}\rvert)$	5
$\mathrm{Min}(\lvert J_{23}\rvert,\lvert J_{13}\rvert)$	0
$\mathrm{Min}(\lvert J_{12}\rvert,\lvert J_{13}\rvert)$	0
$\mathrm{Min}(\lvert J_{12}\rvert,\lvert J_{23}\rvert)$	1
$A(J_{12})$	0
$A(J_{23})$	3
$A(J_{13})$	0
$\Delta G(H_1,H_2)$	1.5
$\Delta G(H_2,H_3)$	6.0
$\Delta G(H_1,H_3)$	-3.3

Figure 5.5 An example of a three-way junction and its 15 feature values.

Specifically, for five-way or higher-order junctions, every pair of neighboring helices and their in-between loop region is considered whether the coaxial stacking is formed or not. In Table 5.2, one can observe a set of 10 feature values used for predicting helical coaxial stacking on a pair of neighboring helices H_i and H_{i+1}. Therefore, for $n > 4$, n

sets of feature values are extracted from an *n*-way junction. Like the features used for three-way and four-way junctions, the loop region length, the maximum number of consecutive adenines and thermodynamic free-energy values for the current pair of neighboring helices ($|J_{i(i+1)}|$, $A(J_{i(i+1)})$, $\Delta G(H_i,H_{i+1})$) are considered, as well as those for the previous pair ($|J_{(i-1)i}|$, $A(J_{(i-1)i})$, $\Delta G(H_{i-1},H_i)$) and the following pair ($|J_{(i+1)(i+2)}|$, $A(J_{(i+1)(i+2)})$, $\Delta G(H_{i+1},H_{i+2})$). Figure 5.6 shows an example of a five-way junction and a set of 10 feature values for pair of helix H_3 and helix H_4.



Feature Name	Feature Value
$\|J_{34}\|$	2
$\|J_{23}\|$	2
$\|J_{45}\|$	3
$\mathrm{Min}(\|J_{23}\|,\|J_{45}\|)$	2
$A(J_{34})$	0
$A(J_{23})$	0
$A(J_{45})$	3
$\Delta G(H_3,H_4)$	6.9
$\Delta G(H_2,H_3)$	6.9
$\Delta G(H_4,H_5)$	6.6

Figure 5.6 An example of a five-way junction and a set of 10 feature values for a pair of helix H_3 and helix H_4.

5.3.3 Random Forests Algorithm

The random forests algorithm was first proposed by Breiman in 2001 [65]. This algorithm employs a number of Classification and Regression Trees (CART, a kind of binary decision tree) which are built, during the training phase, with the features introduced in the previous section. In the testing phase, a test sample will be classified based on the majority votes from all decision trees. The detail of random forest algorithm is explained below.

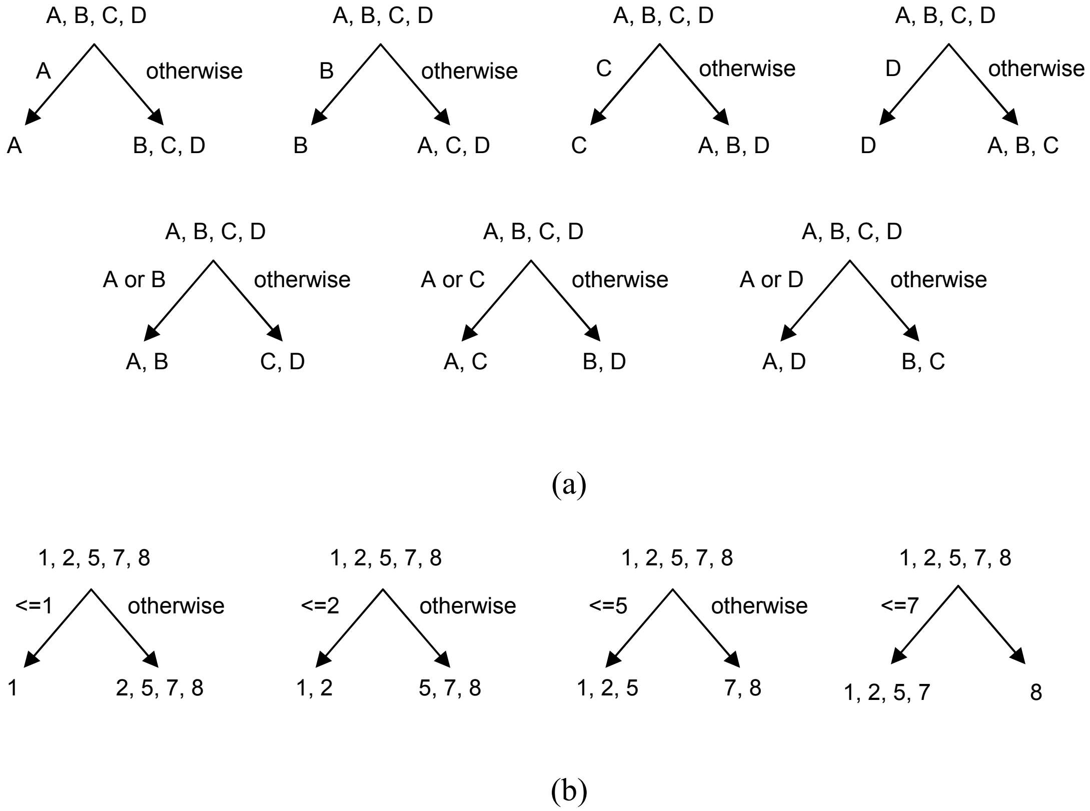


Figure 5.7 (a) There are 7 possible splits if a categorical attribute contains 4 different categories: A, B, C and D. (b) There are 4 possible splits if a numerical attribute contains 5 different numerical values: 1, 2, 5, 7 and 8.

Suppose the number of training records is N. Randomly pick records N times with replacement (repeatedly picking the same record is allowed). According to $(1\text{-}1/N)^N = 1/e = 0.368$ when N approaches infinity, about 63.2% of training records will be picked to grow each decision tree. The remaining set, with approximately 36.8% of training data, will be used for error rate estimation. Suppose the number of attributes in each training record is M. When splitting each node, $\sqrt{M}$ attributes are randomly picked. Each possible split from all picked attributes is examined and the best split, determined by the gini impurity measure, among them is used to split the node. Suppose an attribute is a categorical variable of n different categories. There are 2^{n-1}-1 possible splits. In Figure 5.7a, there are 7 possible splits shown if a categorical attribute contains 4 different categories: A, B, C and D. Suppose an attribute is a numerical variable of n different values. There are n-1 possible splits. An example in Figure 5.7b shows that there are 4 possible splits if a numerical attribute contains 5 different numerical values: 1, 2, 5, 7 and 8.

Suppose there are m classes in a node t which is going to split to t_L and t_R. The gini impurity measure for t is:

$$g(t) = 1 - \sum_{i=1}^{m} f_i^2 \tag{5.3}$$

where f_i is the fraction of class i among all training records in t. If there is only one class in node t, then $g(t)$ is zero; otherwise, $g(t)$ is greater than zero.

The equations to determine the best split are

$$\Delta g(s,t) = g(t) - P_L g(t_L) - P_R g(t_R) \tag{5.4}$$

and

$$S^* \leftarrow \arg\max_s(\Delta g(s,t)) \tag{5.5}$$

where s is a split, P_L and P_R are the proportion of training records assigned to t_L and t_R respectively according to s. S^* is the best split among all possible splits. Among all possible splits from randomly picked attributes, the best split is the split with the greatest decrease in the gini impurity measure.

Table 5.3 is an example of a sample dataset used to demonstrate how to train a random forests classifier. In this dataset, sixteen records and four attributes are contained, including **RNA_Type**, **Minimum_Loopsize**, **Protein_interaction** and **Family_Type**. **Coaxail_Stacking** is the attribute/label to be predicted. To build a CART binary decision tree, suppose that six records are randomly picked as shown in Table 5.4. Since the square root of four (attributes) is two, two attributes need to be randomly picked to split the root node *t*. Suppose two attributes **Minimum_Loopsize** and **Family_Type** are randomly picked. The best split among all splits from **Minimum_Loopsize** and **Family_Type** will be used to split node *t* to child nodes t_L and t_R as shown in Figure 5.8. In attribute **Coaxial_Stacking**, there are 4 records with "Yes" and 2 records with "No" in root node *t*. According to Equation (5.3), the gini impurity measure for *t* is $1\text{-}(4/6)^2\text{-}(2/6)^2\text{=}0.444$. For attribute **Minimum_Loopsize**, because it contains two different numerical values which are 1 and 2, there is only one possible split. For attribute **Family_Type**, there are three possible splits as it contains three different categorical values which are A, B and C. Therefore, the best split will be determined among these four possible splits which are s_1, s_2, s_3 and s_4. In Figures 5.9, 5.10, 5.11 and 5.12, according to Equation (5.4), the calculation of each split's decrease in the gini impurity measure is shown. Through Equation (5.5), the best split is s_3 since it has the greatest decrease in the gini impurity measure. In Figure 5.13, the root node *t* is split to the left child node and the right child node by attribute **Family_Type**'s value B. Furthermore, on the left child node, since the attribute **Coaxial_Stacking**'s values of both records are Yes, no further split is necessary and this left child node is labeled as Yes. That is, in the testing phase, any test record falling into this left child node will be classified/labeled/predicted as Yes to their **Coaxial_Stacking** attribute by the decision tree. On the right child node, the training dataset is shrunk to four records (two Yes and two No) in the attribute **Coaxial_Stacking**. Therefore, the split is needed for the right child node as shown in Figure 5.13.

Table 5.3 An Example of a Sample Dataset Used to Demonstrate How to Train a Random Forests Classifier

RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
tRNA	1	No	A	Yes
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No
23S rRNA	2	Yes	B	Yes
23S rRNA	2	Yes	C	Yes
23S rRNA	1	No	B	Yes
23S rRNA	2	Yes	A	Yes
16S rRNA	1	Yes	C	No
tRNA	1	Yes	B	Yes
16S rRNA	2	No	A	No
tRNA	2	No	C	Yes
tRNA	2	No	B	Yes
23S rRNA	2	No	B	Yes
tRNA	2	Yes	A	No
tRNA	2	Yes	B	No
23S rRNA	2	No	A	No

Table 5.4 Six Records Randomly Picked from Table 5.3

RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
tRNA	1	No	A	Yes
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No
23S rRNA	2	Yes	B	Yes
23S rRNA	2	Yes	C	Yes
23S rRNA	1	No	B	Yes

Minimum_Loopsize	Family_Type	Coaxial_Stacking
1	A	Yes
1	A	No
2	C	No
2	B	Yes
2	C	Yes
1	B	Yes

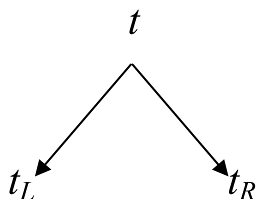
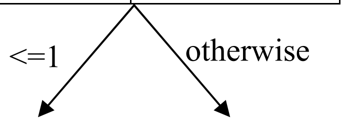


Figure 5.8 The best split among all splits from Minimum_Loopsize and Family_Type will be used to split node t to child nodes t_L and t_R.

Minimum_Loopsize	Coaxial_Stacking
1	Yes
1	No
2	No
2	Yes
2	Yes
1	Yes



Minimum_Loopsize	Coaxial_Stacking
1	Yes
1	No
1	Yes

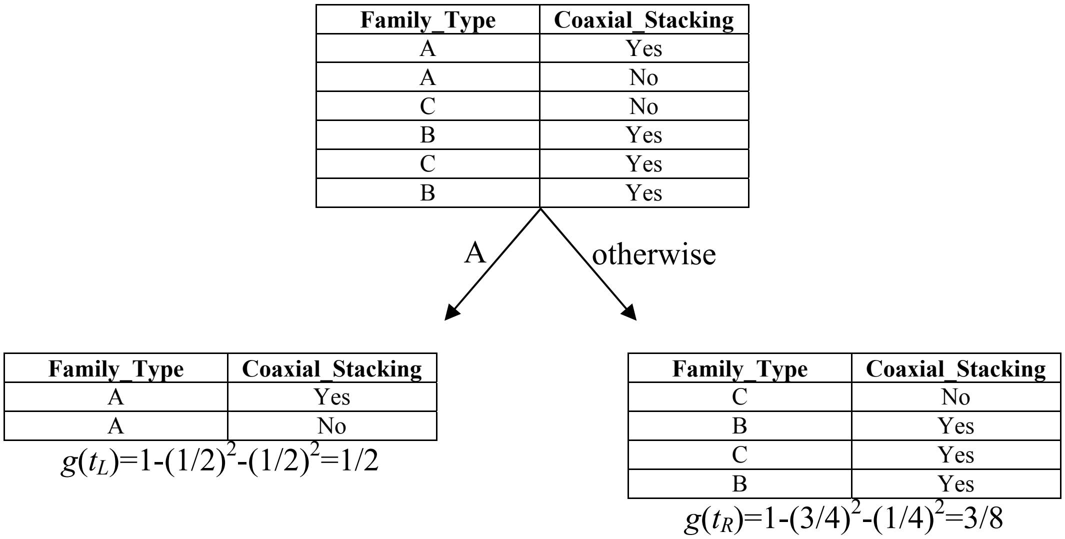
$g(t_L)$=1-(2/3)^2-(1/3)^2=4/9

Minimum_Loopsize	Coaxial_Stacking
2	No
2	Yes
2	Yes

$g(t_R)$=1-(2/3)^2-(1/3)^2=4/9

$\Delta g(s_1,t)$=0.44-(3/6)(4/9)-(3/6)(4/9)=0

Figure 5.9 The decrease in the gini impurity measure for the split s_1.



Family_Type	Coaxial_Stacking
A	Yes
A	No
C	No
B	Yes
C	Yes
B	Yes

Family_Type	Coaxial_Stacking
A	Yes
A	No

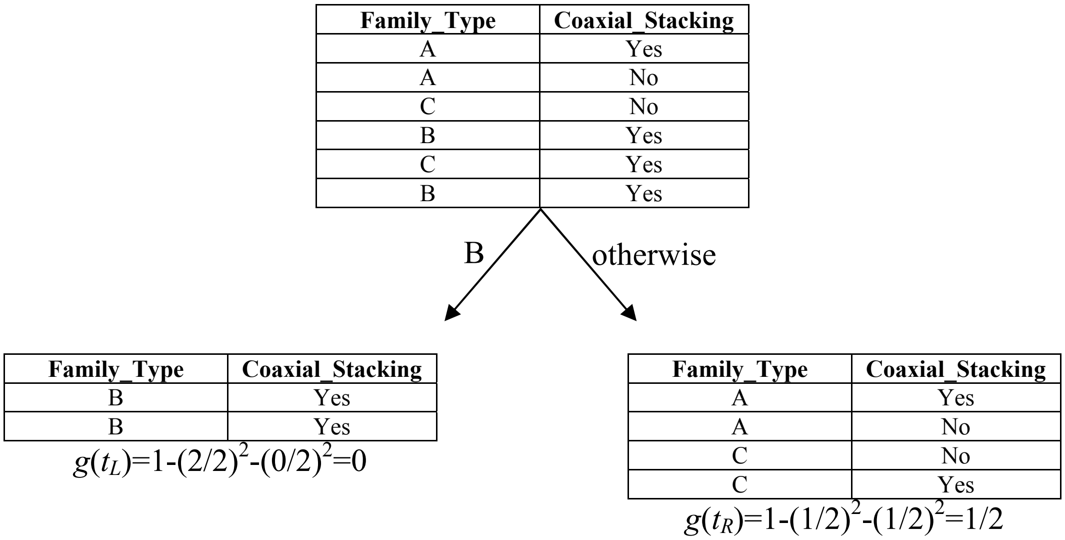
$g(t_L)$=1-(1/2)^2-(1/2)^2=1/2

Family_Type	Coaxial_Stacking
C	No
B	Yes
C	Yes
B	Yes

$g(t_R)$=1-(3/4)^2-(1/4)^2=3/8

$$\Delta g(s_2,t)=0.44-(2/6)(1/2)-(4/6)(3/8)=0.028$$

Figure 5.10 The decrease in the gini impurity measure for the split s_2.



Family_Type	Coaxial_Stacking
A	Yes
A	No
C	No
B	Yes
C	Yes
B	Yes

Family_Type	Coaxial_Stacking
B	Yes
B	Yes

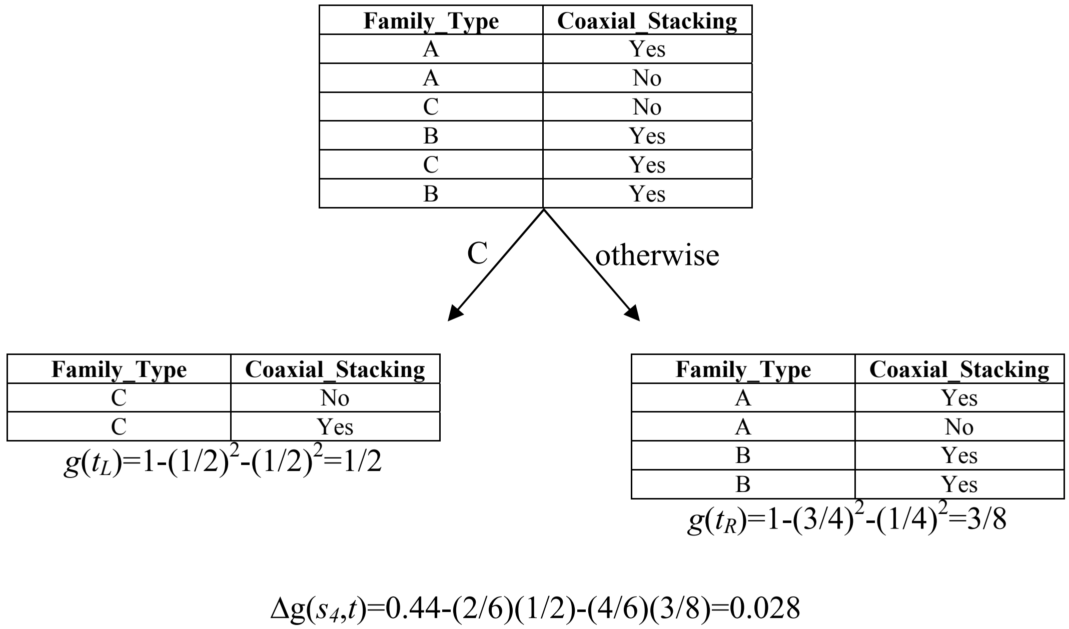
$g(t_L)$=1-(2/2)^2-(0/2)^2=0

Family_Type	Coaxial_Stacking
A	Yes
A	No
C	No
C	Yes

$g(t_R)$=1-(1/2)^2-(1/2)^2=1/2

$$\Delta g(s_3,t)=0.44-(2/6)(0)-(4/6)(1/2)=0.11$$

Figure 5.11 The decrease in the gini impurity measure for the split s_3.



Family_Type	Coaxial_Stacking
A	Yes
A	No
C	No
B	Yes
C	Yes
B	Yes

Family_Type	Coaxial_Stacking
C	No
C	Yes

$g(t_L)$=1-(1/2)^2-(1/2)^2=1/2

Family_Type	Coaxial_Stacking
A	Yes
A	No
B	Yes
B	Yes

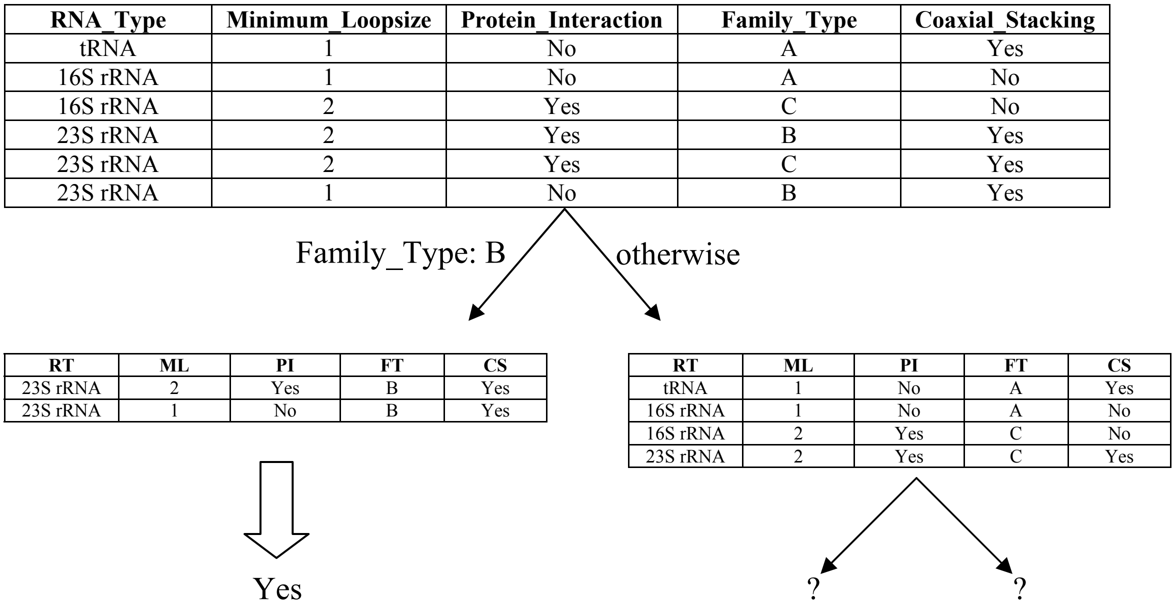
$g(t_R)$=1-(3/4)^2-(1/4)^2=3/8

Δg(s_4,t)=0.44-(2/6)(1/2)-(4/6)(3/8)=0.028

Figure 5.12 The decrease in the gini impurity measure for the split s_4.

$\Delta g(s_1,t)=0$
$\Delta g(s_2,t)=0.028$
$\Delta g(s_3,t)=0.11$
$\Delta g(s_4,t)=0.028$

s_3 is the best split among four possible splits



RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
tRNA	1	No	A	Yes
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No
23S rRNA	2	Yes	B	Yes
23S rRNA	2	Yes	C	Yes
23S rRNA	1	No	B	Yes

RT	ML	PI	FT	CS
23S rRNA	2	Yes	B	Yes
23S rRNA	1	No	B	Yes

RT	ML	PI	FT	CS
tRNA	1	No	A	Yes
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No
23S rRNA	2	Yes	C	Yes

Figure 5.13 The result of the split on the root node *t*.

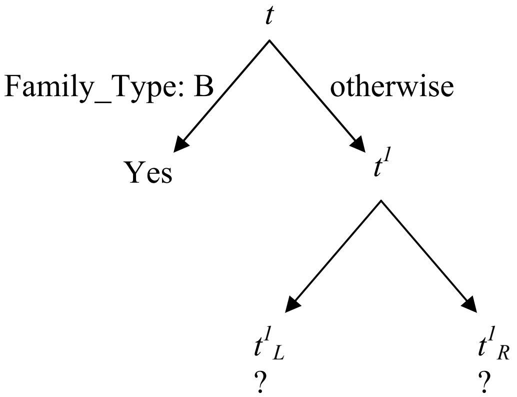


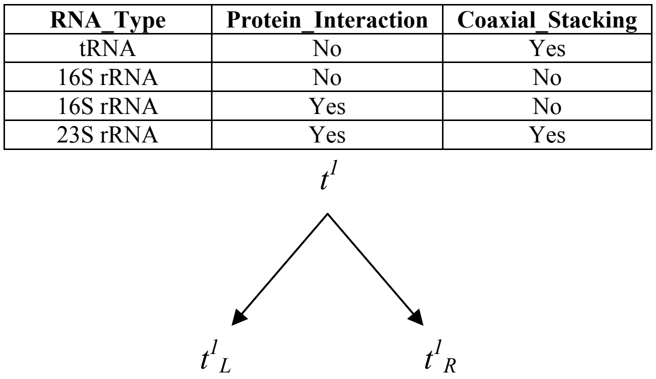
Figure 5.14 The current status of the CART binary decision tree.

The current status of the CART binary decision tree is shown as Figure 5.14. The training records used to split t^1 are listed as Table 5.5. Since the square root of four (attributes) is two, two attributes need to be randomly picked to split the node t^1. Suppose two attributes **RNA_Type** and **Protein_Interaction** are randomly picked. The best split among all splits from **RNA_Type** and **Protein_Interaction** will be used to split node t^1 to node t^1_L and t^1_R as shown in Figure 5.15. In attribute **Coaxial_Stacking**, there are 2 records with "Yes" and 2 records with "No" in node t^1. According to Equation (5.3), the gini impurity measure for t^1 is $1\text{-}(2/4)^2\text{-}(2/4)^2=0.5$. For attribute **RNA_Type**, there are three possible splits as it contains three different categorical values: tRNA, 16S rRNA and 23S rRNA. For attribute **Protein_Interaction**, there is only one possible split as it contains two different categorical values which are Yes and No. Therefore, the best split will be determined among the following four possible splits: s^1_1, s^1_2, s^1_3 and s^1_4. In Figures 5.16, 5.17, 5.18 and 5.19, according to Equation (5.4), the calculation of each split's decrease in the gini impurity measure is shown. Referring to Equation (5.5), the best split is s^1_3 as it has the greatest decrease in the gini impurity measure. In Figure 5.20, the node t^1 is split to the left child node and the right child node by attribute **RNA_Type**'s value 16S rRNA. On the left child node, since the attribute **Coaxial_Stacking**'s values of both records are No, no further split is necessary and this left child node is labeled as No. On the right child node, since the attribute **Coaxial_Stacking**'s values of both records are Yes, no further split is necessary and this right child node is labeled as Yes.

Therefore, in Figure 5.21, one complete CART binary decision tree is grown and trained by the randomly picked six records as shown in Table 5.4. In Figures 5.22 and 5.23, another two CART binary decision trees are grown and trained by two training sets of records randomly picked from the sample datasets of Table 5.3. In Figure 5.24, an unlabeled testing record is shown and three separate decisions/classifications are made for the testing record by three CART binary decision trees. Since the majority vote is Yes (two for Yes and one for No), the random forests' final decision/classification for the testing record is Yes.

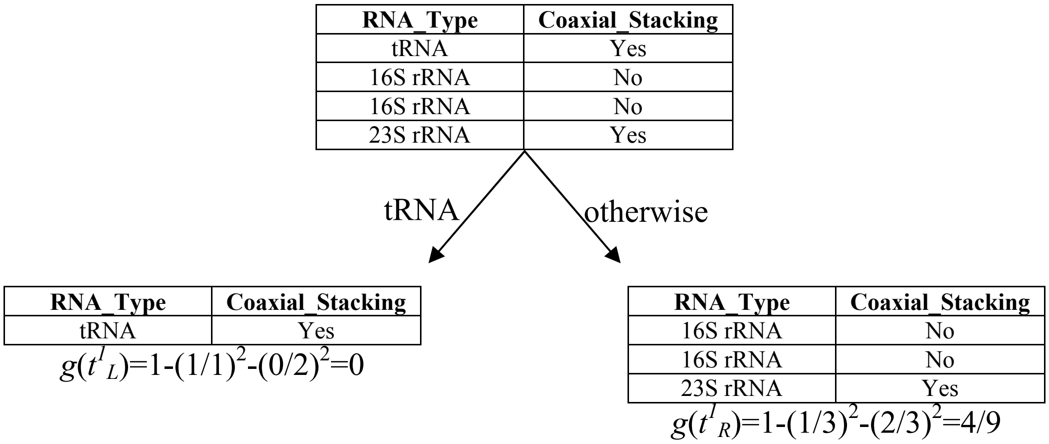
Table 5.5 The Training Records Used to Split t^1

RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
tRNA	1	No	A	Yes
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No
23S rRNA	2	Yes	C	Yes



RNA_Type	Protein_Interaction	Coaxial_Stacking
tRNA	No	Yes
16S rRNA	No	No
16S rRNA	Yes	No
23S rRNA	Yes	Yes

Figure 5.15 The best split among all of the splits from RNA_Type and Protein_Interaction will be used to split node t^1 to child nodes t^1_L and t^1_R.



RNA_Type	Coaxial_Stacking
tRNA	Yes
16S rRNA	No
16S rRNA	No
23S rRNA	Yes

RNA_Type	Coaxial_Stacking
tRNA	Yes

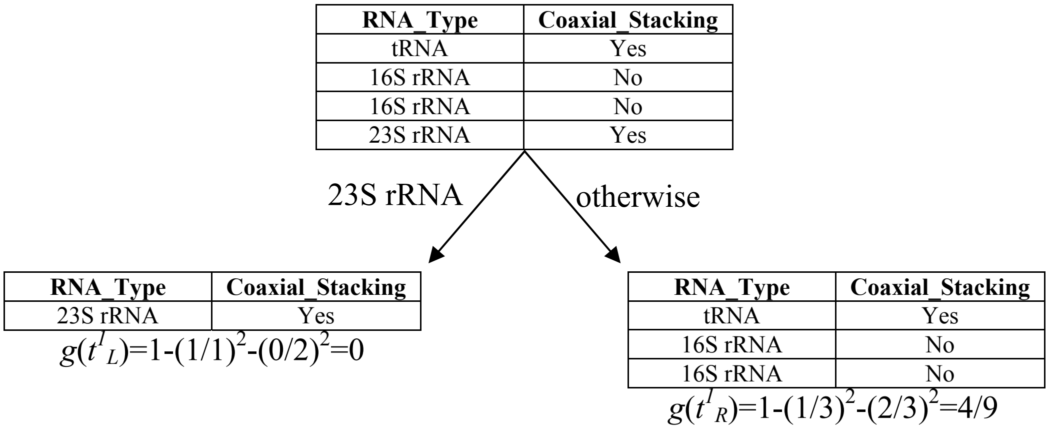
$g(t^1{}_L)$=1-(1/1)^2-(0/2)^2=0

RNA_Type	Coaxial_Stacking
16S rRNA	No
16S rRNA	No
23S rRNA	Yes

$g(t^1{}_R)$=1-(1/3)^2-(2/3)^2=4/9

Δg($s^1{}_1$,t^1)=0.5-(1/4)(0)-(3/4)(4/9)=0.167

Figure 5.16 The decrease in the gini impurity measure for the split $s^1{}_1$.



RNA_Type	Coaxial_Stacking
tRNA	Yes
16S rRNA	No
16S rRNA	No
23S rRNA	Yes

RNA_Type	Coaxial_Stacking
23S rRNA	Yes

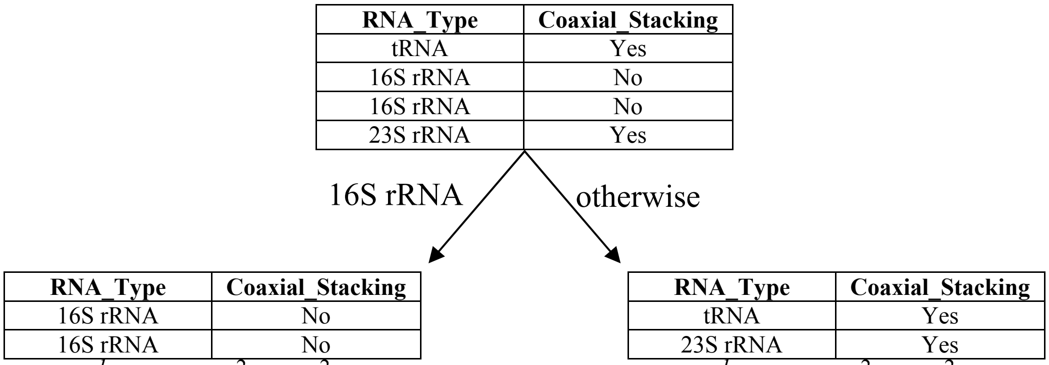
$g(t^1{}_L)$=1-(1/1)^2-(0/2)^2=0

RNA_Type	Coaxial_Stacking
tRNA	Yes
16S rRNA	No
16S rRNA	No

$g(t^1{}_R)$=1-(1/3)^2-(2/3)^2=4/9

Δg($s^1{}_2$,t^1)=0.5-(1/4)(0)-(3/4)(4/9)=0.167

Figure 5.17 The decrease in the gini impurity measure for the split $s^1{}_2$.



RNA_Type	Coaxial_Stacking
tRNA	Yes
16S rRNA	No
16S rRNA	No
23S rRNA	Yes

RNA_Type	Coaxial_Stacking
16S rRNA	No
16S rRNA	No

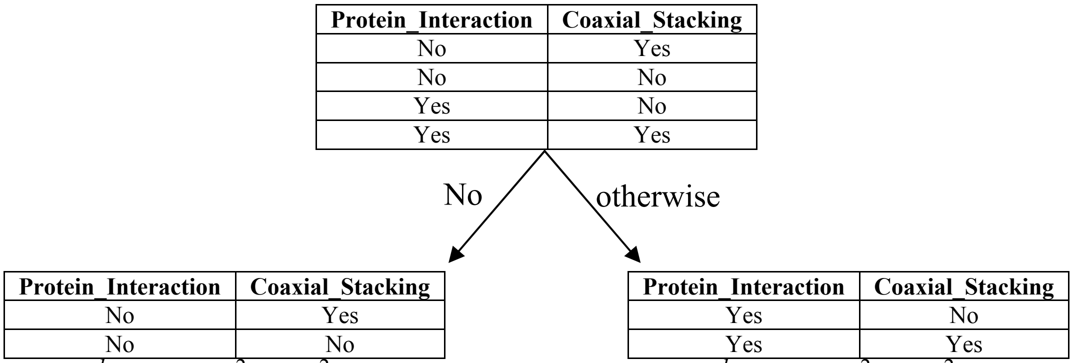
$g(t^1_L)$=1-(0/2)^2-(2/2)^2=0

RNA_Type	Coaxial_Stacking
tRNA	Yes
23S rRNA	Yes

$g(t^1_R)$=1-(2/2)^2-(0/0)^2=0

Δg(s^1_3,t^1)=0.5-(2/4)(0)-(2/4)(0)=0.5

Figure 5.18 The decrease in the gini impurity measure for the split s^1_3.



Protein_Interaction	Coaxial_Stacking
No	Yes
No	No
Yes	No
Yes	Yes

Protein_Interaction	Coaxial_Stacking
No	Yes
No	No

$g(t^1_L)$=1-(1/2)^2-(1/2)^2=1/2

Protein_Interaction	Coaxial_Stacking
Yes	No
Yes	Yes

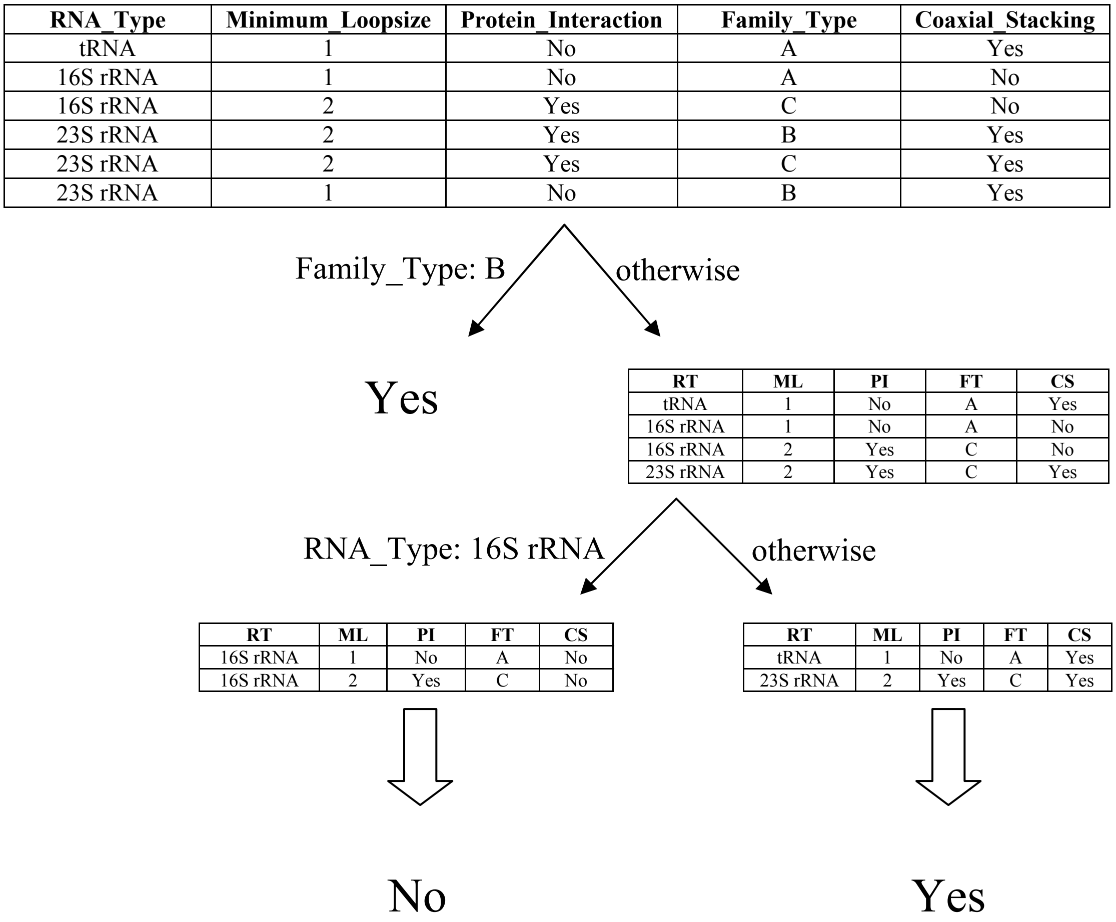
$g(t^1_R)$=1-(1/2)^2-(1/2)^2=1/2

Δg(s^1_4,t^1)=0.5-(2/4)(1/2)-(2/4)(1/2)=0

Figure 5.19 The decrease in the gini impurity measure for the split s^1_4.

$\Delta g(s^1_1,t^1)=0.167$
$\Delta g(s^1_2,t^1)=0.167$
$\Delta g(s^1_3,t^1)=0.5$
$\Delta g(s^1_4,t^1)=0$

s^1_3 is the best split among four possible splits



RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
tRNA	1	No	A	Yes
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No
23S rRNA	2	Yes	B	Yes
23S rRNA	2	Yes	C	Yes
23S rRNA	1	No	B	Yes

RT	ML	PI	FT	CS
tRNA	1	No	A	Yes
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No
23S rRNA	2	Yes	C	Yes

RT	ML	PI	FT	CS
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No

RT	ML	PI	FT	CS
tRNA	1	No	A	Yes
23S rRNA	2	Yes	C	Yes

Figure 5.20 The result of the split on the node t^1.

RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
tRNA	1	No	A	Yes
16S rRNA	1	No	A	No
16S rRNA	2	Yes	C	No
23S rRNA	2	Yes	B	Yes
23S rRNA	2	Yes	C	Yes
23S rRNA	1	No	B	Yes

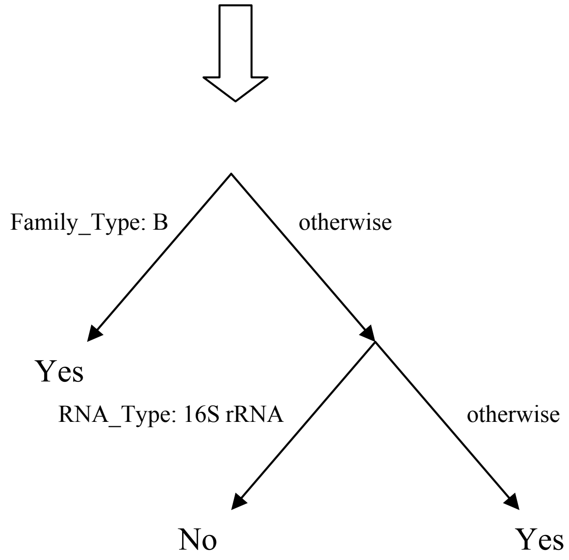


Figure 5.21 The complete CART binary decision tree is grown and trained by the randomly picked six records as shown in Table 5.4.

RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
23S rRNA	1	No	B	Yes
23S rRNA	2	Yes	A	Yes
16S rRNA	1	Yes	C	No
16S rRNA	2	No	A	No
tRNA	1	Yes	B	Yes

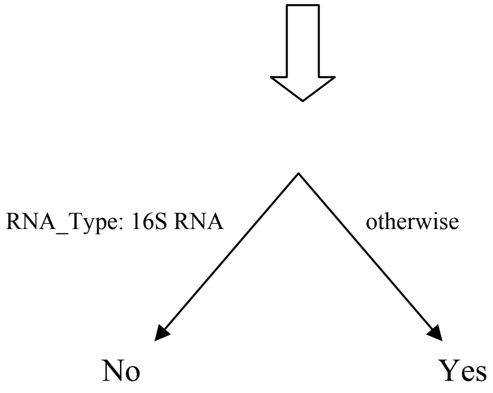


Figure 5.22 A complete CART binary decision tree is grown and trained by five records randomly picked from Table 5.3.

RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
tRNA	2	No	C	Yes
tRNA	2	No	B	Yes
23S rRNA	2	No	B	Yes
tRNA	2	Yes	A	No
tRNA	2	Yes	B	No
23S rRNA	2	No	A	No
16S rRNA	2	No	A	No

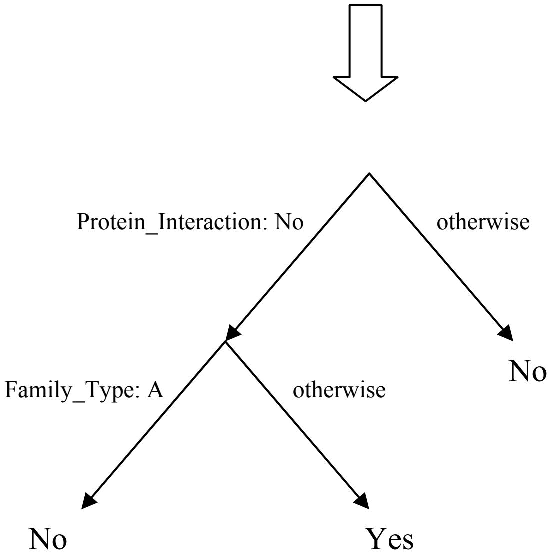


Figure 5.23 A complete CART binary decision tree is grown and trained by seven records randomly picked from Table 5.3.

RNA_Type	Minimum_Loopsize	Protein_Interaction	Family_Type	Coaxial_Stacking
16S rRNA	2	No	B	?

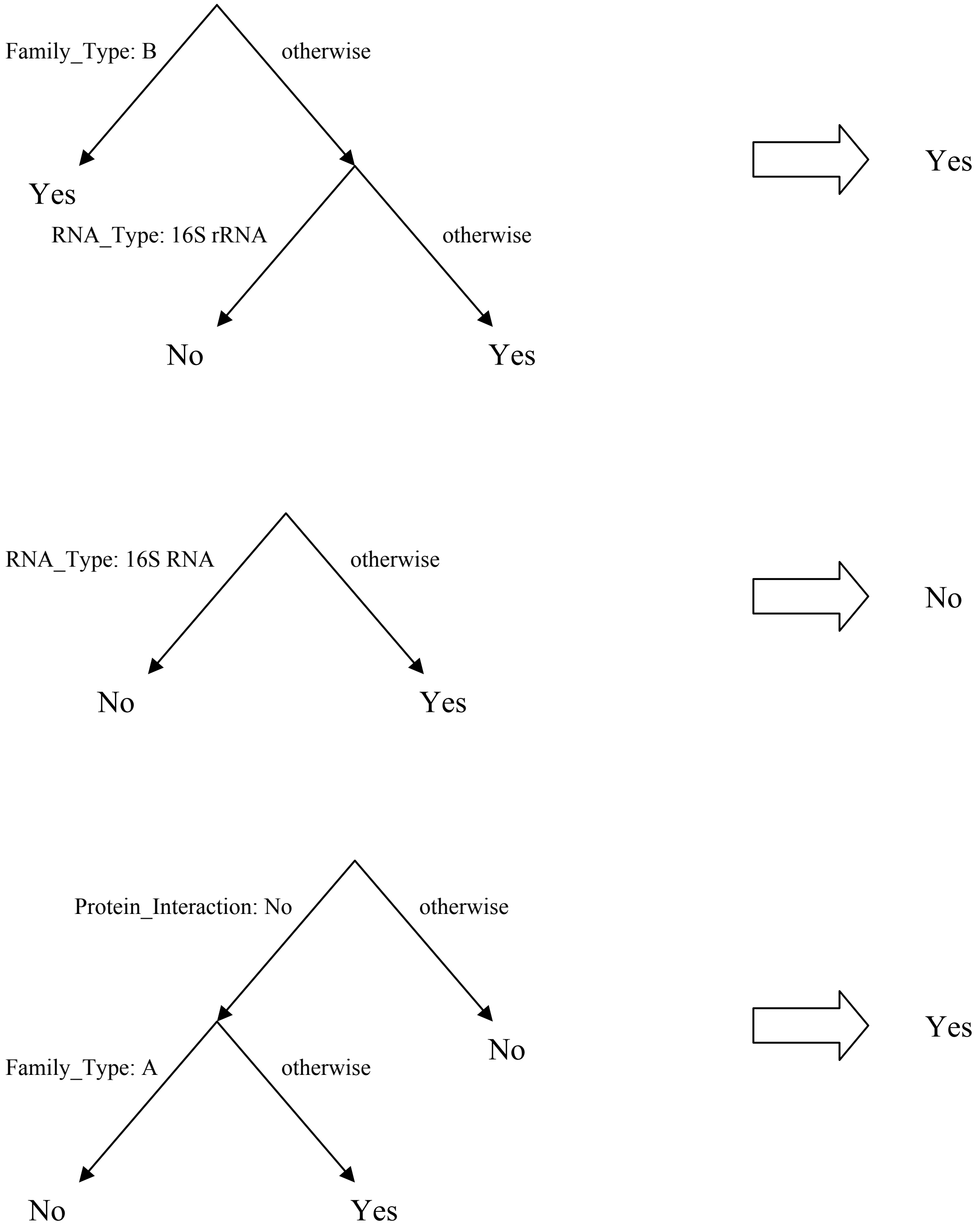


Figure 5.24 There is an unlabeled testing record and three separate decisions are made for the testing record by three CART binary decision trees. Since the majority vote is Yes (two for Yes and one for No), the random forests' final decision/classification for the testing record is Yes.

5.4 Experiments and Performance Evaluation

To ensure the accuracy and the performance of the proposed method, the experiments and the performance evaluations including the comparison with other works are reported in this section. In Table 5.6, all numbers regarding datasets used for the experiments are listed. For three-way junctions, there are 110 junctions which include 4 different coaxial stacking classes (H_1H_2, H_2H_3, H_1H_3 and no coaxial stacking) and 3 families (A, B and C). For four-way junctions, there are 65 junctions that include 7 different coaxial stacking classes (H_1H_2, H_2H_3, H_3H_4, H_1H_4, H_1H_2-H_3H_4, H_2H_3-H_1H_4 and no coaxial stacking) and 9 families (H, cH, cL, cK, π, cW, ψ, cX, and X). For a higher-order junction (5-way or more), due to the lack in the data collection, all higher-order junctions are combined (41 junctions in total), the common features locally extracted and the coaxial stacking locally predicted (two classes: positive and negative). The programs of the training and prediction phases for all experiments are implemented on the R software for statistical computing with the random forest package installed [66].

5.4.1 Results of Experiments

To avoid bias, 75 repeats of 10-fold cross validation and 200 trees grown for each random forests classifier are used for all experiments. In one single example of 10-fold cross validation, the entire dataset is randomly separated into 10 groups. Each group of data takes turns as testing data while the remaining 9 groups are used as random forests classifier's training data. Therefore, there are 10 different sets of random forests generated with 200 trees each in one single of 10-fold cross validation. Finally, the average of those 750 accuracy percentages is reported for each experiment.

In Table 5.7, for three-way and four-way junctions, the accuracies of the coaxial stacking prediction without any family information are 81% and 77% respectively. When considering the family information, the accuracies of the coaxial stacking prediction are 83% and 87% for the three-way and four-way junctions respectively. On the other hand, the accuracies of the junction family prediction without any coaxial stacking information are 85% and 74% for the three-way and four-way junctions respectively. When considering the coaxial stacking information for the three-way and four-way junctions, the accuracies of the junction family prediction are 86% and 81% respectively.

Table 5.6 The Numbers Regarding Datasets Used for the Experiments

The order of junctions	**Number of junctions**	**Number of coaxial stacking classes**	**Number of families**
3-way	110	4	3
4-way	65	7	9
5~10-way	41	2	-

Table 5.7 The Performance of the Coaxial Stacking and Junction Family Predictions for Three-way and Four-way Junctions

		3-way junction	**4-way junction**
Coaxial stacking prediction	**Family is unknown**	81%	77%
	Family is known	83%	87%
Junction family prediction	**Coaxial stacking is unknown**	85%	74%
	Coaxial stacking is known	86%	81%

Table 5.8 The Performance of the Coaxial Stacking Prediction for the Higher-order Junctions

		5~10-way junction
Coaxial stacking prediction	**Accuracy**	60%
	Positive predictive value	76%

Table 5.9 The Performance of the Coaxial Stacking Prediction in Two Steps for the Three-way and Four-way Junctions

	3-way junction	**4-way junction**
Step1: Junction family prediction	85%	73%
Step 2: Coaxial stacking prediction	82%	80%

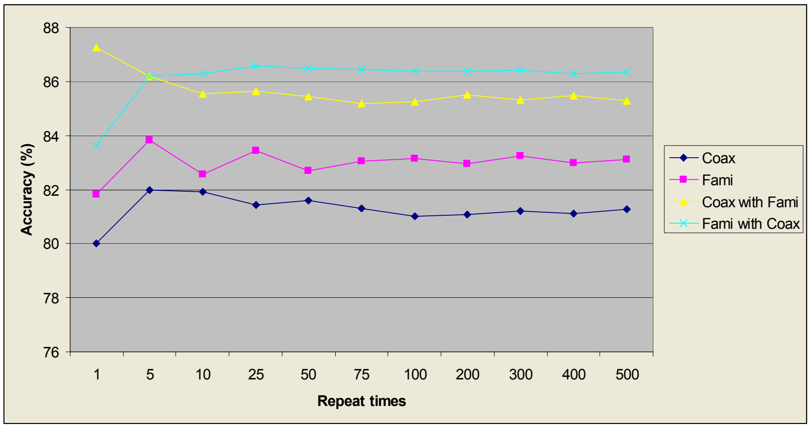
Table 5.10 The Performance of the Junction Family Prediction in Two Steps for the Three-way and Four-way Junctions

	3-way junction	**4-way junction**
Step1: Coaxial stacking prediction	82%	77%
Step 2: Junction family prediction	86%	71%

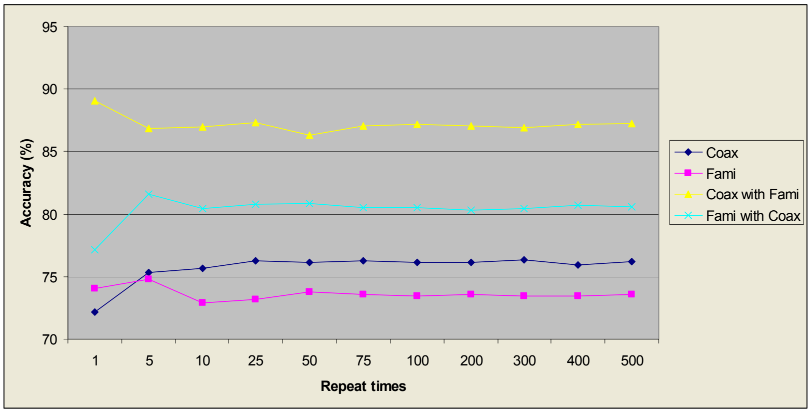
In Table 5.8, the accuracy and the positive predictive value of the coaxial stacking prediction for the higher-order junctions (five-way to ten-way junctions) are shown to be 60% and 76%, respectively. In this experiment of Table 5.8, 211 sets of common features are extracted from the entire higher-order junctions and applied to the experiment of 10-fold cross validation. To improve the prediction performance, 590 sets of common features extracted from the three-way and four-way junctions are included in the training dataset during the experiment.

The accuracy of the coaxial stacking prediction is improved when the junction family information is included in the feature sets. However, the methods to manually collect the additional information such as junction topology might be expensive, impractical and time consuming. Therefore, the random forests prediction for junction topology provides an alternative. Here, a new prediction procedure for coaxial stacking in two steps is proposed. In the first step, the type of junction family is predicted. In the second step, the junction family information predicted in the previous step is added into the feature sets. Thus the prediction of coaxial stacking is performed by using the feature sets with their new contents. In Tables 5.9 and 5.10, the results of this proposed two-step prediction procedure for three-way and four-way junctions are shown in two different orders. In Table 5.9, the junction family is predicted in the first step and then the coaxial stacking is predicted in the second step. On the other hand, in Table 5.10, the coaxial stacking is predicted first followed by the junction family prediction.

To avoid bias, 75 repeats of 10-fold cross validation and 200 trees grown for each random forests classifier are used as parameters for all experiments. The choices of these two parameters were analyzed and optimized by testing several values. In Figure 5.25, all prediction performances of both coaxial stacking and junction family with a fixed number of trees and a series of different repeat times are shown on three-way and four-way junctions. Figure 5.26 shows all prediction performances of both coaxial stacking and junction family with a fixed number of repeat times and a series of different numbers of trees on three-way and four-way junctions. Through these figures, the convergence of the prediction accuracy is found and the smallest parameter values producing approximately the same prediction accuracy are selected.

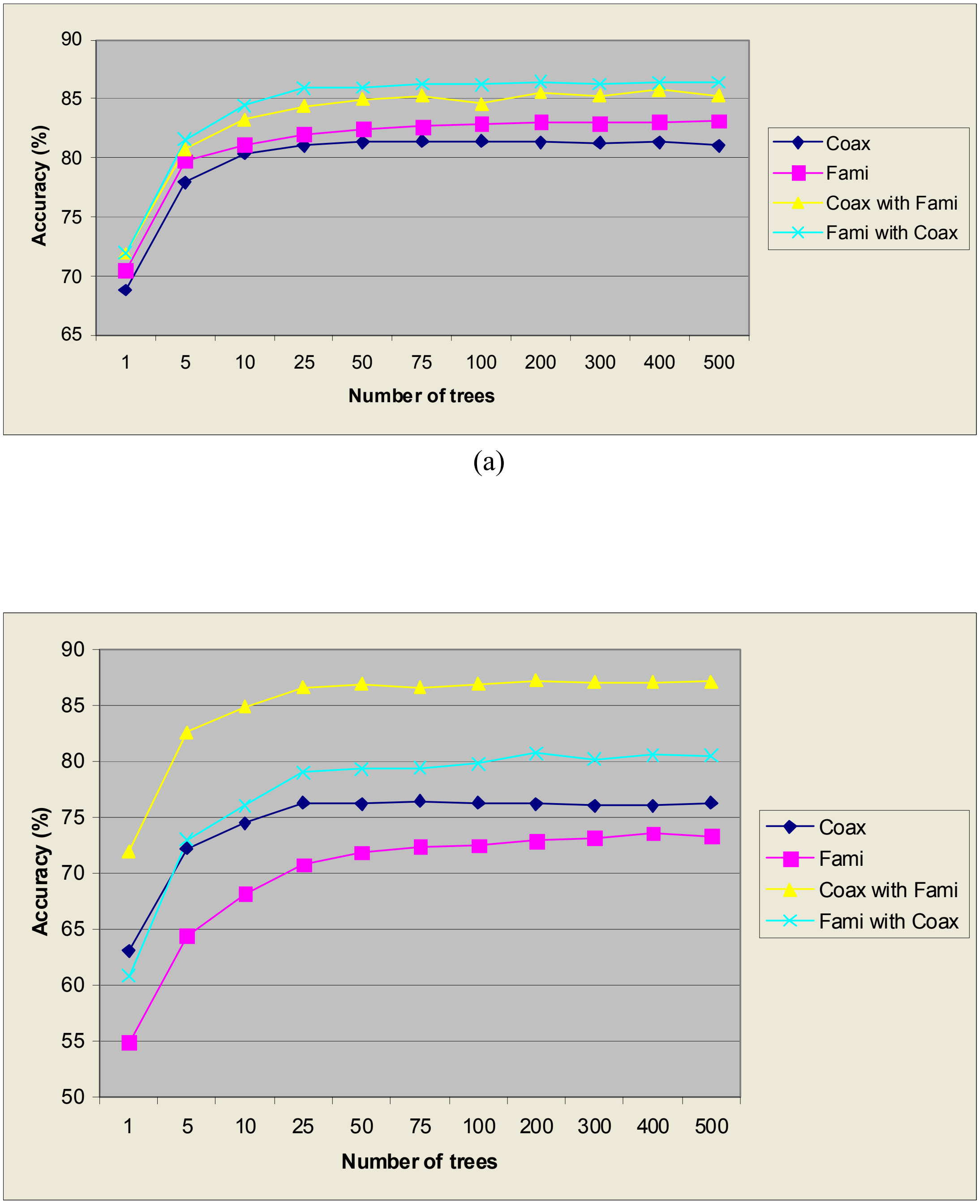


(a)



(b)

Figure 5.25 The prediction performances of coaxial stacking and junction family with a fixed number of trees and a series of different repeat times. (a) The polygonal graph for three-way junctions. (b) The polygonal graph for four-way junctions.



(a)

(b)

Figure 5.26 The prediction performances of coaxial stacking and junction family with a fixed number of repeat times and a series of different numbers of trees. (a) The polygonal graph for three-way junctions. (b) The polygonal graph for four-way junctions.

Table 5.11 The Prediction Result Comparisons on Three-way Junctions of Unsolved RNA Structures

RNA type	Domain	Lescoute & Westhof [58]	Tyagi & Mathews [62]	The proposed RF classifier
VS ribozyme	II-III-VI	H2H3, Family A	H2H3	H1H2, Family C
VS ribozyme	III-IV-V	H1H2, Family C	H2H3	H1H2, Family C
DiGIR1	P3-P8-P15	H2H3, Family C	H1H3	H2H3, Family C
U4U6	I-II-III	H1H2, Family B	None	H1H3, Family C
HCV	IIIo-IIIabc-IIId	H1H2, Family C or H2H3, Family A	H2H3	H1H2, Family C
RNase P	P5-P5.1-P7	H1H2, Family A	None	H1H3, Family C

5.4.2 The Comparison with Other Publications

By using free energy minimization, Tyagi and Mathews predicted coaxial stacking between pairs of consecutive helices with one or none intervening mismatch loops [62]. It is very difficult to formulate a direct comparison with the approach of Tyagi and Mathews as their prediction is restricted to the pair of consecutive helices with one or none intervening mismatch loops. The method proposed in this dissertation is able to predict the coaxial stacking with any size of mismatch loops. The definition of junctions differs as well. Tyagi and Mathews consider helical stems as those formed by at least one base pair, while helical stems as those formed by at least two base pair are considered here. The disagreement on coaxial stacking configuration for the same junction exists between their dataset and the one used in this dissertation.

To make a consistent comparison, the junctions with agreement on the definition and on the coaxial stacking configuration between Tyagi and Mathews' dataset and ours are used as the testing dataset. The remainder of our dataset is used as the training dataset. For three-way junctions, there are 91 training junctions and 20 testing junctions. For four-way junctions, there are 49 training junctions and 27 testing junctions. The random forests classifier shows an accuracy rate of 80% on three-way junctions and an accuracy rate of 92.59% on four-way junctions while Tyagi and Mathews have 30% and 70.37% respectively.

Lescoute and Westhof predicted the topology and coaxial stacking configuration on three-way junctions of RNA whose structures have not yet been solved at atomic resolution [58]. The RNAs include the 'Varkud' satellite ribozyme (VS), the *Didymium* group I-like intron ribozyme (DiGIR1), a three-way junction formed between the U4 and U6 RNAs in the spliceosome (U4U6), the hepatitis C virus (HCV), and the recently solved RNase P. Tyagi and Mathews also presented their coaxial stacking predictions on the same junctions. Table 5.11 lists the prediction results of Lescoute and Westhof, Tyagi and Mathews, and the proposed random forests classifier as well as the RNA type and its domain. Figure 5.27 shows the result of three different accuracy comparisons with Tyagi and Mathews.

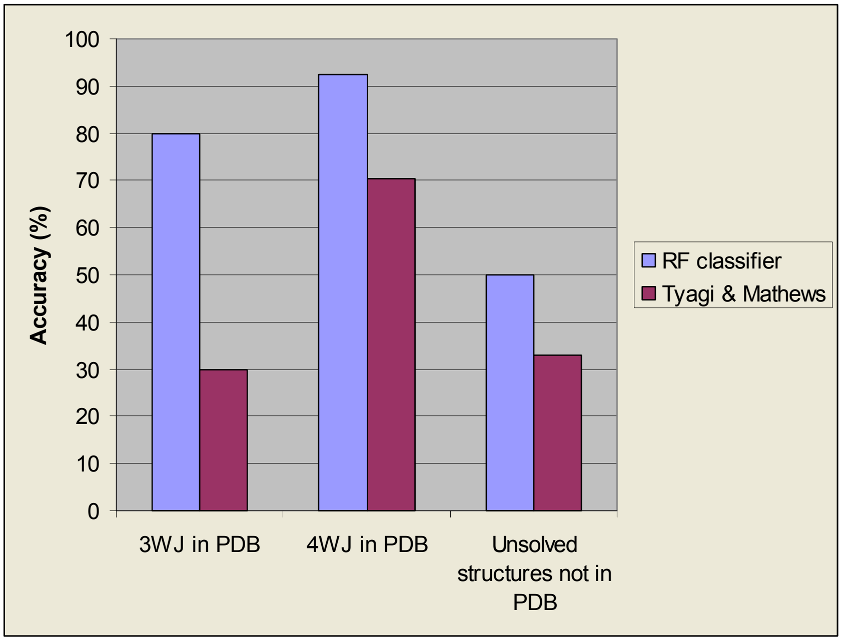


Figure 5.27 The bar chart of three different accuracy comparisons with the work of Tyagi and Mathews [62].

5.5 Feature Ranking Analysis

The feature sets used in this study are extracted from the size and sequence loop, as well as the base pair configuration of junctions. In order to improve the prediction, the significance analysis of each feature used to predict coaxial stacking and junction topology on three-way and four-way junctions is reported in this section. Two different feature ranking algorithms are used to analyze the features. The details of algorithms and the analyzing results are described below.

5.5.1 Feature Ranking by Single Feature Accuracy

The concept of feature ranking by single feature accuracy is quite simple. The feature ranking could be acquired by following step 1 – 4.

1. Each time, leaving only one feature within the feature set.
2. Performing the 10-fold cross validation with parameters as 75 repeat times and 200 trees for each random forest.
3. The average accuracy of 75 times of 10-fold cross validations is recorded.
4. Go to Step 1 until every feature is chosen.

Obviously, after the above procedure, each feature is associated with a percentage of accuracy, thus permitting all of the features to be ranked by percentages of accuracy. If a feature is ranked on top, that feature's contribution to the prediction accuracy is more significant than the others. Tables 5.12, 5.13, 5.14 and 5.15 list the rankings, by accuracy, of features on predictions of the coaxial stacking and the junction family on three-way and four-way junctions. To estimate the optimal size of feature set, another analysis is performed. The number of features within the feature set in the order of ranking by accuracy from the best significant feature to the worst significant feature is accumulated.

In Figures 5.28 and 5.29, the polygonal graphs show the trend of prediction accuracies from the feature sets containing anywhere from one feature to the full set of features.

Table 5.12 The Feature Ranking by Accuracy of the Coaxial Stacking Prediction on Three-way Junctions

Features	Rank by accuracy
$\|J_{23}\|$	52.71
$\Delta G(H_2,H_3)$	50.045
$Med(\|J_{12}\|,\|J_{23}\|,\|J_{31}\|)$	49.765
$Max(\|J_{12}\|,\|J_{23}\|,\|J_{31}\|)$	49.7
$\Delta G(H_3,H_1)$	48.295
$A(J_{12})$	48.25
$\|J_{31}\|$	46.915
$\Delta G(H_1,H_2)$	44.74
$Min(\|J_{23}\|,\|J_{31}\|)$	44.685
$Min(\|J_{12}\|,\|J_{31}\|)$	43.87
$A(J_{31})$	41.07
$\|J_{12}\|$	40.78
$A(J_{23})$	39.52
$Min(\|J_{12}\|,\|J_{23}\|,\|J_{31}\|)$	39.2
$Min(\|J_{23}\|,\|J_{12}\|)$	34.455

Table 5.13 The Feature Ranking by Accuracy of the Junction Family Prediction on Three-way Junctions

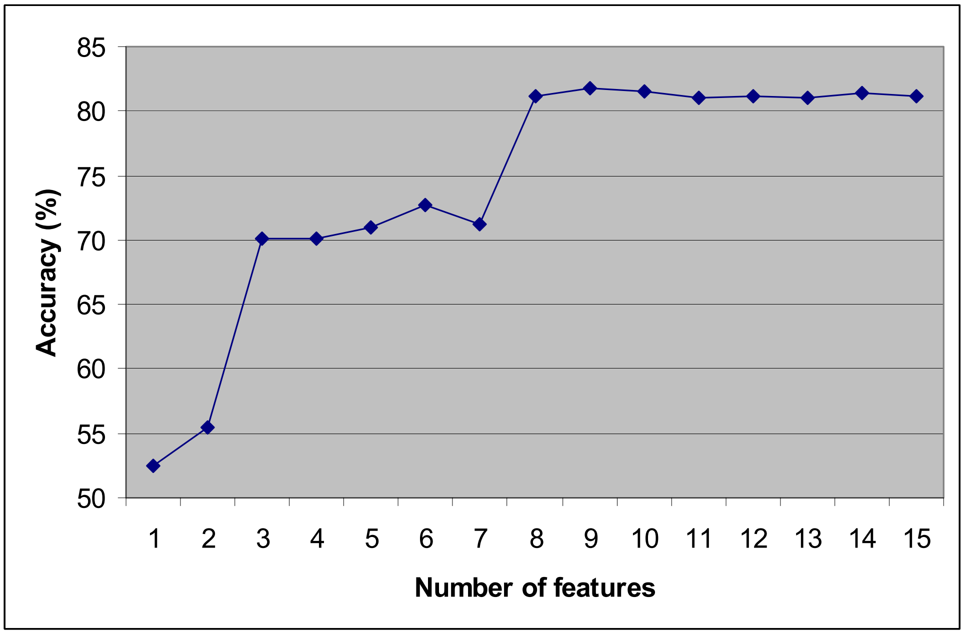
Features	Rank by accuracy
$\Delta G(H_1,H_2)$	57.325
$A(J_{23})$	55.2
$Max(\|J_{12}\|,\|J_{23}\|,\|J_{31}\|)$	54.74
$\|J_{23}\|$	51.92
$Min(\|J_{23}\|,\|J_{12}\|)$	50.3
$\|J_{31}\|$	49.13
$Min(\|J_{23}\|,\|J_{31}\|)$	48.105
$\|J_{12}\|$	46.625
$Min(\|J_{23}\|,\|J_{31}\|)$	46.565
$\Delta G(H_2,H_3)$	46.26
$Min(\|J_{12}\|,\|J_{23}\|,\|J_{31}\|)$	45.895
$Med(\|J_{12}\|,\|J_{23}\|,\|J_{31}\|)$	45.84
$\Delta G(H_3,H_1)$	45.54
$A(J_{31})$	44.14
$A(J_{12})$	43.365

Table 5.14 The Feature Ranking by Accuracy of the Coaxial Stacking Prediction on Four-way Junctions

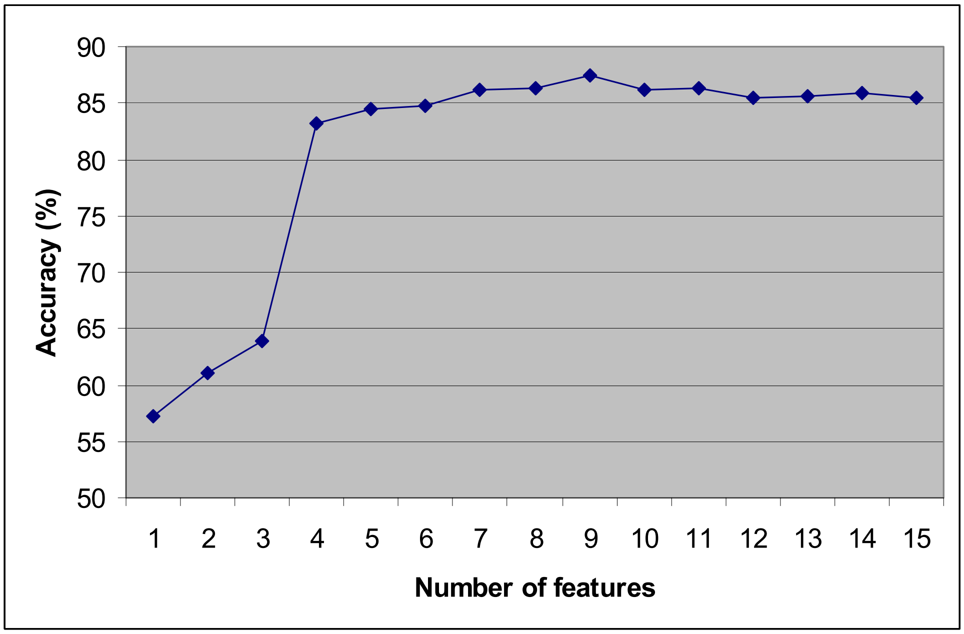
Features	Rank by accuracy
$\Delta G(H_3,H_4)$	62.325
$Min(J_{12},J_{34})$	61.33
$\|J_{34}\|$	61.025
$\Delta G(H_1,H_4)$	60.07
$Med_{max}(\|J_{12}\|,\|J_{23}\|,\|J_{34}\|,\|J_{41}\|)$	56.94
$\|J_{41}\|$	55.975
$Min(J_{23},J_{41})$	55.75
$Max(\|J_{12}\|,\|J_{23}\|,\|J_{34}\|,\|J_{41}\|)$	55.29
$Med_{min}(\|J_{12}\|,\|J_{23}\|,\|J_{34}\|,\|J_{41}\|)$	54.35
$A(J_{41})$	52.665
$A(J_{23})$	50.5
$Min(\|J_{12}\|,\|J_{23}\|,\|J_{34}\|,\|J_{41}\|)$	49.345
$A(J_{34})$	47.305
$\Delta G(H_1,H_2)$	45.055
$\|J_{12}\|$	44.26
$\|J_{23}\|$	42.745
$\Delta G(H_2,H_3)$	39.77
$A(J_{12})$	39.325

Table 5.15 The Feature Ranking by Accuracy of the Junction Family Prediction on Four-way Junctions

Features	Rank by accuracy
$\|J_{34}\|$	47.33
$Max(\|J_{12}\|,\|J_{23}\|,\|J_{34}\|,\|J_{41}\|)$	44.88
$Min(J_{12},J_{34})$	42.905
$\Delta G(H_3,H_4)$	39.14
$\Delta G(H_2,H_3)$	38.2
$\Delta G(H_1,H_4)$	36.82
$\|J_{41}\|$	36.77
$Med_{max}(\|J_{12}\|,\|J_{23}\|,\|J_{34}\|,\|J_{41}\|)$	33.38
$\|J_{23}\|$	30.6
$\Delta G(H_1,H_2)$	27.955
$Min(J_{23},J_{41})$	24.865
$A(J_{12})$	20.6
$\|J_{12}\|$	20.37
$Med_{min}(\|J_{12}\|,\|J_{23}\|,\|J_{34}\|,\|J_{41}\|)$	20.02
$Min(\|J_{12}\|,\|J_{23}\|,\|J_{34}\|,\|J_{41}\|)$	19.96
$A(J_{34})$	17.66
$A(J_{23})$	15.73
$A(J_{41})$	13.4

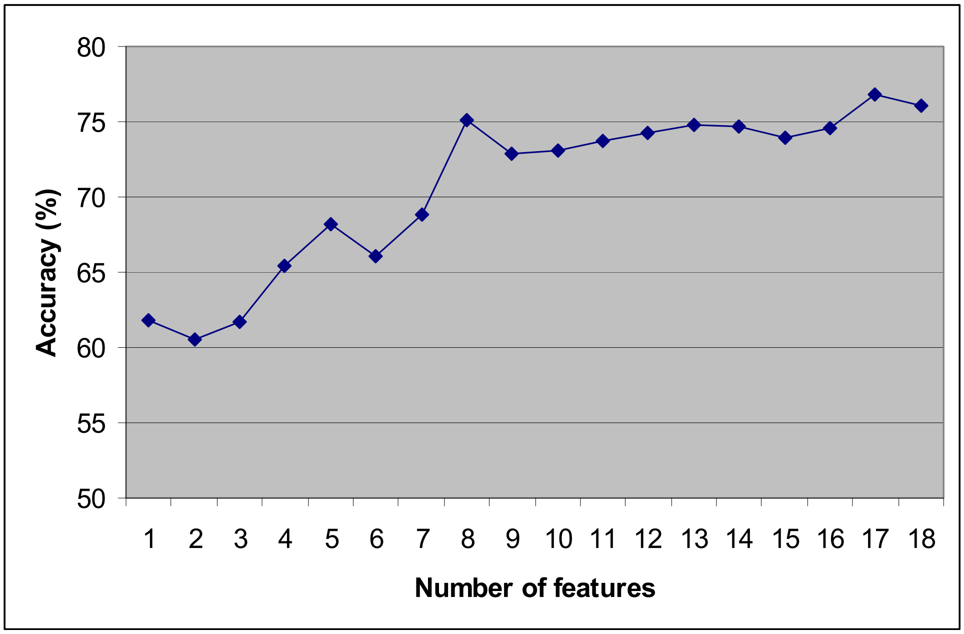


(a)

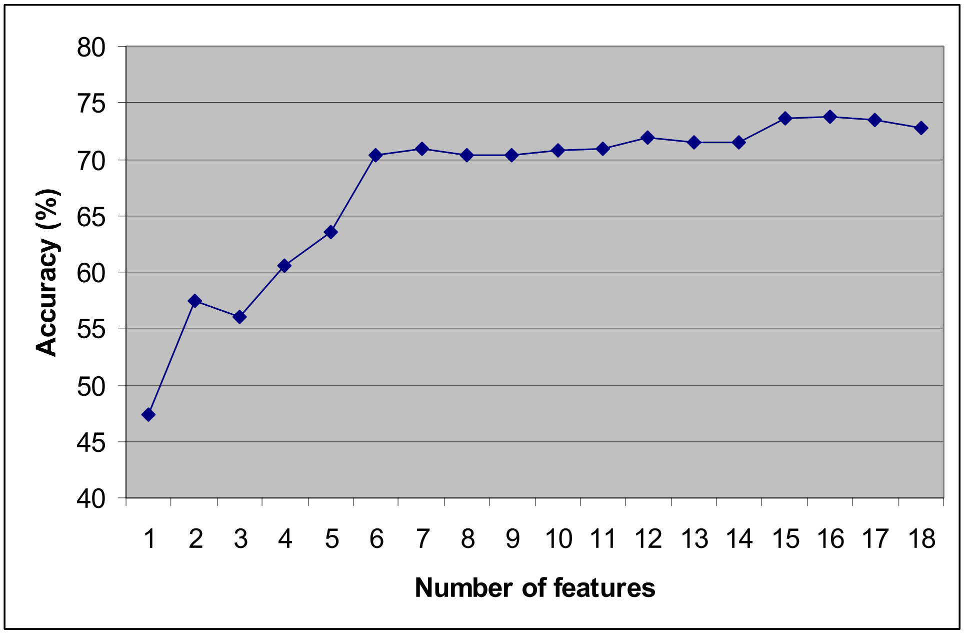


(b)

Figure 5.28 Accumulating the number of features within the feature set in the order of ranking from the best significant feature to the worst significant feature. The prediction accuracies from the feature sets containing one feature to the full set of features is plotted. (a) The polygonal graph of the coaxial stacking prediction for three-way junctions. (b) The polygonal graph of the junction family prediction for three-way junctions.



(a)



(b)

Figure 5.29 Accumulating the number of features within the feature set in the order of ranking from the best significant feature to the worst significant feature. The prediction accuracies from the feature sets containing one feature to the full set of features is plotted. (a) The polygonal graph of the coaxial stacking prediction for four-way junctions. (b) The polygonal graph of the junction family prediction for four-way junctions.

5.5.2 Feature Ranking by the Gini Impurity Measure of Random Forests Algorithm

As described in Section 5.3.3, in the random forests algorithm, every node split is associated with a maximum decrease in the gini impurity measure and one specific feature value. When the training of the random forests classifier is complete, the average of the maximum decrease in the gini impurity measure, taken from every node split, for each feature can be calculated.

For example, suppose that there are four features (F_1, F_2, F_3 and F_4) in a training set. After the training phase of the random forests algorithm, there are two trees, Tree 1 and Tree 2, generated for the classifier as shown in Figure 5.30. For Tree 1, initially four variables, $SUM^1{}_{F1}$, $SUM^1{}_{F2}$, $SUM^1{}_{F3}$ and $SUM^1{}_{F4}$ are all zeros. After Tree 1 is generated, the variables become as follows:

$$SUM^1{}_{F1} = \Delta g(s_{F1}, t) + \Delta g(s_{F1}^{1,L}, t_L^1)$$

$$SUM^1{}_{F2} = \Delta g(s_{F2}^{1,R}, t_R^1)$$

$$SUM^1{}_{F3} = 0$$

$$SUM^1{}_{F4} = 0$$

Similarly, for Tree 2, initially four variables, $SUM^2{}_{F1}$, $SUM^2{}_{F2}$, $SUM^2{}_{F3}$ and $SUM^2{}_{F4}$ are all zeros. After Tree 2 is generated, these variables become as follows:

$$SUM^2{}_{F1} = \Delta g(s_{F1}, t) + \Delta g(s_{F1}^{1,R}, t_R^1)$$

$$SUM^2{}_{F2} = 0$$

$$SUM^2{}_{F3} = 0$$

$$SUM^2{}_{F4} = \Delta g(s_{F4}^{1,L}, t_L^1)$$

Finally, the average of each feature is calculated as follows:

$$AVG_{Fi} = (SUM^1{}_{Fi} + SUM^2{}_{Fi}) / 2, \text{ where } i \text{ is from 1 to 4.}$$

Each feature can be ranked by its own AVG_{Fi}. If a feature is ranked on top, that feature could contribute more decrease in the gini impurity measure than others ranked lower. Tables 5.16, 5.17, 5.18 and 5.19 list the rankings, by the averages of decrease in the gini impurity measure, of the features on predictions of the coaxial stacking and the junction family on three-way and four-way junctions. To avoid bias, the averages of decrease in the gini impurity measure of features are obtained from their random forests classifier with 100,000 trees. To estimate the optimal size of the feature set, another analysis is performed. The number of features within the feature set in the order of the ranking by the averages of decrease in the gini impurity measure from the best significant feature to the worst significant feature is accumulated. In Figures 5.31 and 5.32, the polygonal graphs show the trend of prediction accuracies from the feature sets containing anywhere from one feature to the full set of features.

5.6 Conclusions

In this chapter, a data mining method is described to predict the configuration of helical coaxial stacks and families (topologies) in RNA three-way to ten-way junctions at the secondary structure level. This method adopts the random forests classifier which is trained by solved RNA tertiary structures. The features are extracted from the secondary structure level of RNA junctions and are used to train the random forests classifier. The overall accuracy of the prediction from the proposed method is about 80% and the performance is comparable with previous work. Furthermore, the features, which are extracted from the junctions and used for prediction, are analyzed for future improvement. In the next chapter, a web server named Junction-Explorer, built by the proposed method, is introduced. Junction-Explorer can identify and locate the junctions on the RNA secondary structure. For each identified RNA junction, the web server is able to predict the presence of coaxial helical stacking and the topology (family) of the junction.

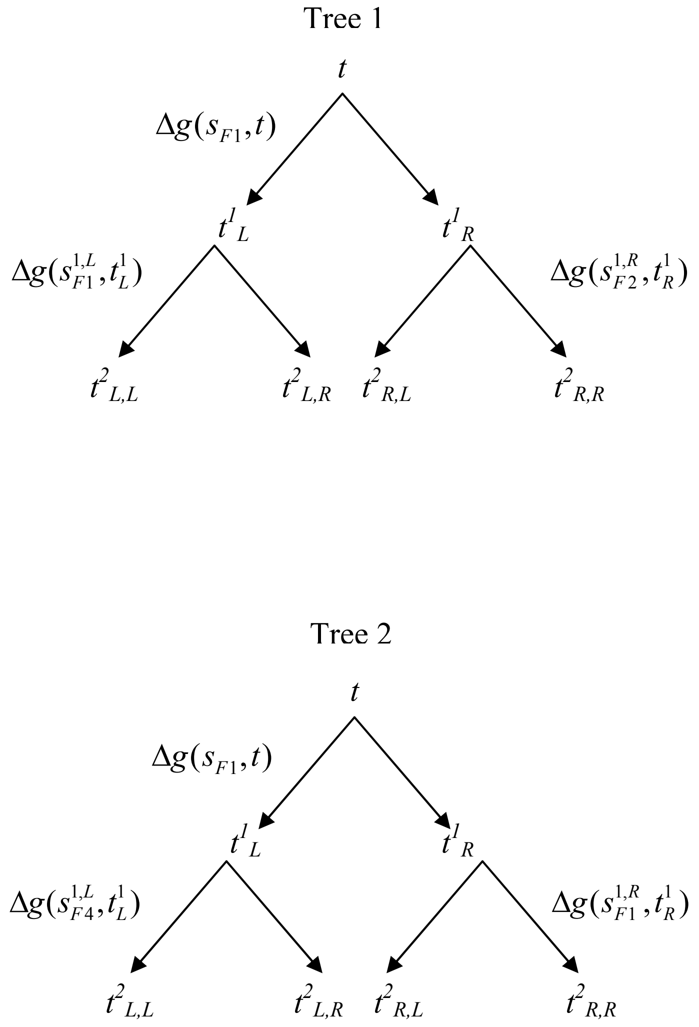


Figure 5.30 Tree 1 and Tree 2 are trained and generated for the random forests classifier.

Table 5.16 The Feature Ranking by Average Δg of the Coaxial Stacking Prediction on Three-way Junctions

Features	Rank by Average Δg
$\lvert J_{23}\rvert$	8.550029
$\lvert J_{31}\rvert$	6.352978
Max($\lvert J_{12}\rvert$,$\lvert J_{23}\rvert$,$\lvert J_{31}\rvert$)	6.124881
ΔG(H_1,H_2)	6.070076
ΔG(H_3,H_1)	5.841098
ΔG(H_2,H_3)	5.711437
$\lvert J_{12}\rvert$	5.430954
Med($\lvert J_{12}\rvert$,$\lvert J_{23}\rvert$,$\lvert J_{31}\rvert$)	5.042766
Min($\lvert J_{12}\rvert$,$\lvert J_{31}\rvert$)	4.256208
A(J_{12})	4.179075
Min($\lvert J_{23}\rvert$,$\lvert J_{12}\rvert$)	4.062099
Min($\lvert J_{23}\rvert$,$\lvert J_{31}\rvert$)	3.912962
Min($\lvert J_{12}\rvert$,$\lvert J_{23}\rvert$,$\lvert J_{31}\rvert$)	3.544635
A(J_{31})	3.279278
A(J_{23})	2.126523

Table 5.17 The Feature Ranking by Average Δg of the Junction Family Prediction on Three-way Junctions

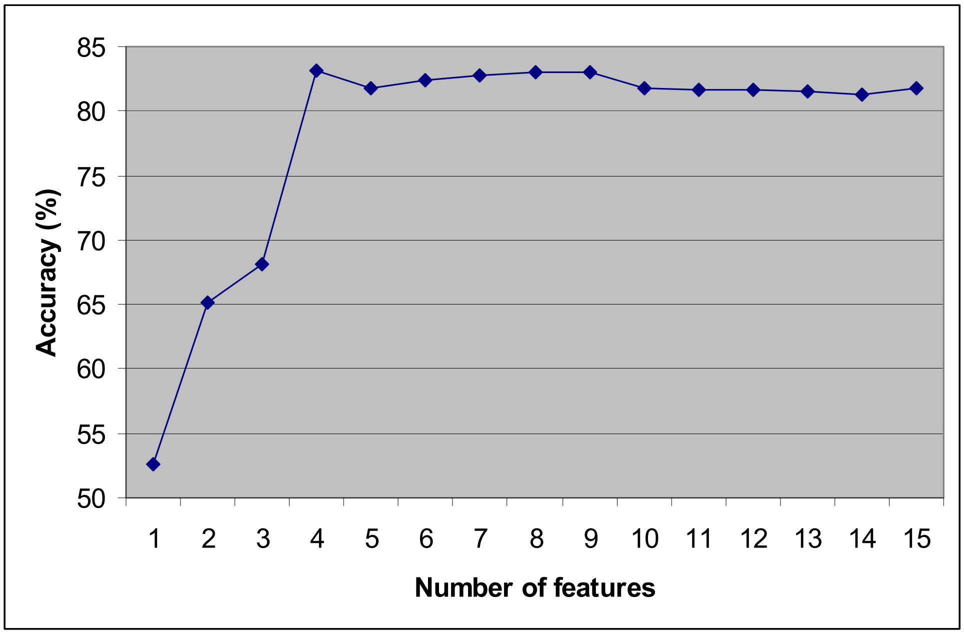
Features	Rank by Average Δg
Max($\lvert J_{12}\rvert$,$\lvert J_{23}\rvert$,$\lvert J_{31}\rvert$)	8.337547
$\lvert J_{23}\rvert$	6.594556
ΔG(H_3,H_1)	5.893594
ΔG(H_1,H_2)	5.51743
$\lvert J_{31}\rvert$	5.032459
ΔG(H_2,H_3)	4.826661
Med($\lvert J_{12}\rvert$,$\lvert J_{23}\rvert$,$\lvert J_{31}\rvert$)	4.480854
$\lvert J_{12}\rvert$	4.321267
Min($\lvert J_{23}\rvert$,$\lvert J_{12}\rvert$)	4.044957
Min($\lvert J_{12}\rvert$,$\lvert J_{31}\rvert$)	3.988266
Min($\lvert J_{23}\rvert$,$\lvert J_{31}\rvert$)	3.936473
Min($\lvert J_{12}\rvert$,$\lvert J_{23}\rvert$,$\lvert J_{31}\rvert$)	3.341332
A(J_{12})	2.388057
A(J_{23})	2.332821
A(J_{31})	1.825746

Table 5.18 The Feature Ranking by Average Δg of the Coaxial Stacking Prediction on Four-way Junctions

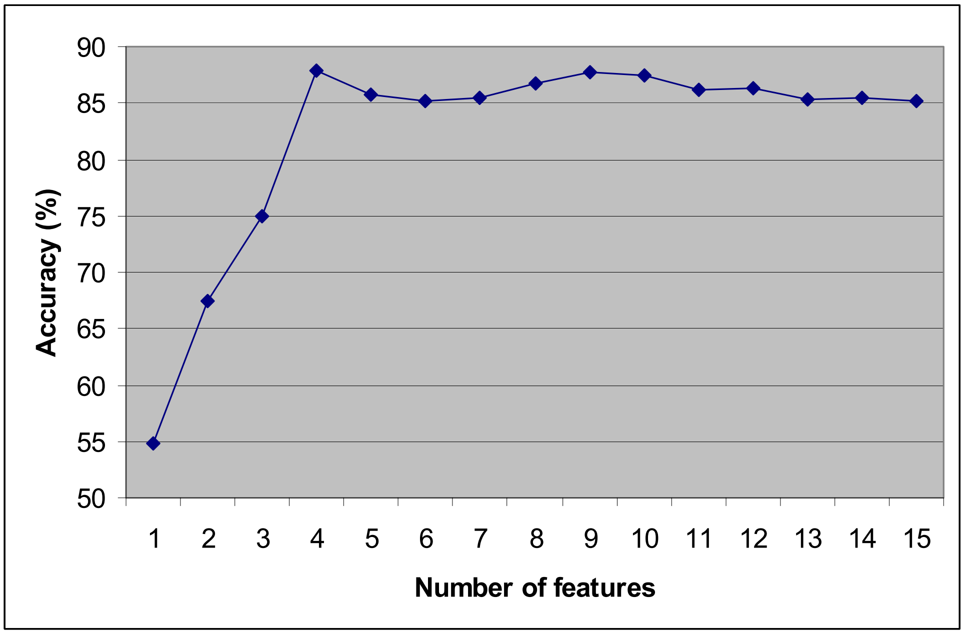
Features	Rank by Average Δg
Max($\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{34}\rvert,\lvert J_{41}\rvert$)	3.9989777
Med_{max}($\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{34}\rvert,\lvert J_{41}\rvert$)	3.7811456
$\lvert J_{34}\rvert$	3.6126635
ΔG(H_1,H_4)	3.341576
$\lvert J_{41}\rvert$	3.2680157
$\lvert J_{23}\rvert$	3.2012284
ΔG(H_2,H_3)	2.9365629
$\lvert J_{12}\rvert$	2.3574117
ΔG(H_1,H_2)	2.3319014
ΔG(H_3,H_4)	2.2672316
Med_{min}($\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{34}\rvert,\lvert J_{41}\rvert$)	2.2476979
A(J_{41})	1.469503
Min($\lvert J_{12}\rvert,\lvert J_{34}\rvert$)	1.4659094
Min($\lvert J_{23}\rvert,\lvert J_{41}\rvert$)	1.1926263
Min($\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{34}\rvert,\lvert J_{41}\rvert$)	1.0316831
A(J_{34})	0.8037794
A(J_{12})	0.7359678
A(J_{23})	0.6250495

Table 5.19 The Feature Ranking by Average Δg of the Junction Family Prediction on Four-way Junctions

Features	Rank by Average Δg
$\lvert J_{34}\rvert$	4.5648662
Max($\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{34}\rvert,\lvert J_{41}\rvert$)	4.4604382
ΔG(H_1,H_4)	3.968873
ΔG(H_1,H_2)	3.5989816
Med_{max}($\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{34}\rvert,\lvert J_{41}\rvert$)	3.5231683
$\lvert J_{41}\rvert$	3.4691833
$\lvert J_{12}\rvert$	3.410726
Min($\lvert J_{12}\rvert,\lvert J_{34}\rvert$)	3.2473648
ΔG(H_2,H_3)	3.1059134
ΔG(H_3,H_4)	2.8694219
$\lvert J_{23}\rvert$	2.3146823
Min($\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{34}\rvert,\lvert J_{41}\rvert$)	2.1493282
Med_{min}($\lvert J_{12}\rvert,\lvert J_{23}\rvert,\lvert J_{34}\rvert,\lvert J_{41}\rvert$)	2.0610557
Min($\lvert J_{23}\rvert,\lvert J_{41}\rvert$)	2.0104957
A(J_{34})	1.3578364
A(J_{41})	1.0358759
A(J_{12})	0.9791774
A(J_{23})	0.6498351

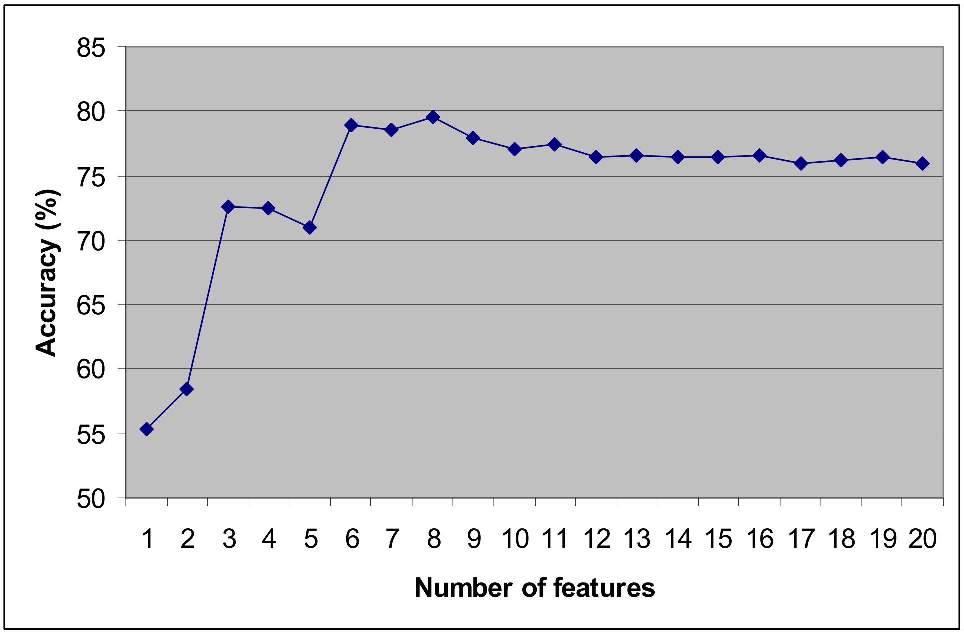


(a)

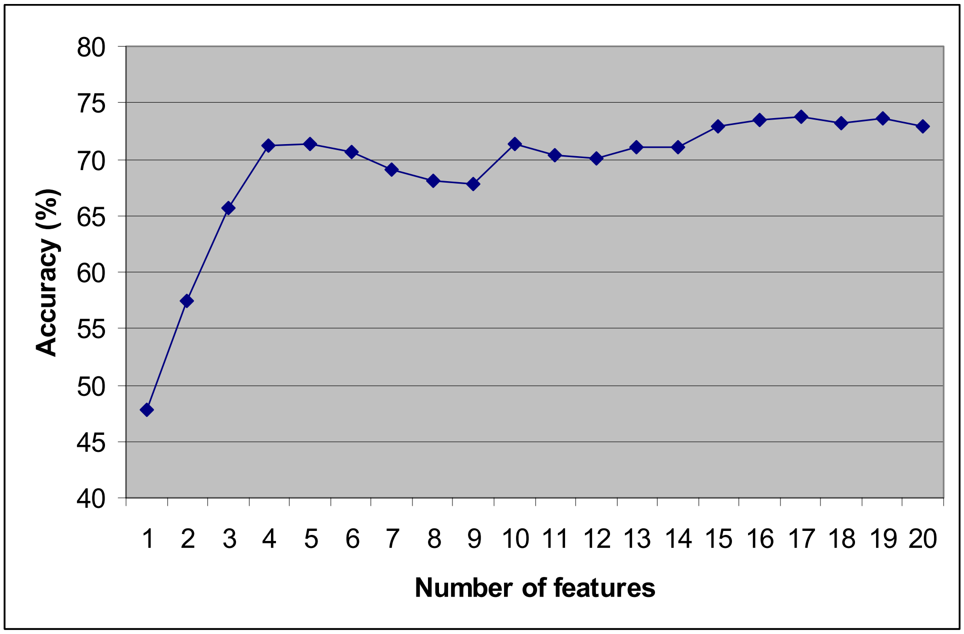


(b)

Figure 5.31 Accumulating the number of features within the feature set in the order of ranking by average Δg from the best significant feature to the worst significant feature. The prediction accuracies from the feature sets containing one feature to the full set of features is plotted. (a) The polygonal graph of the coaxial stacking prediction for three-way junctions. (b) The polygonal graph of the junction family prediction for three-way junctions.



(a)



(b)

Figure 5.32 Accumulating the number of features within the feature set in the order of ranking by average Δg from the best significant feature to the worst significant feature. The prediction accuracies from the feature sets containing one feature to the full set of features is plotted. (a) The polygonal graph of the coaxial stacking prediction for four-way junctions. (b) The polygonal graph of the junction family prediction for four-way junctions.

CHAPTER 6

JUNCTION-EXPLORER

6.1 Overview

As mentioned in the previous chapter, the RNA junctions are important structural elements of three or more helices in the organization of the global structure of RNA molecules. A common motif among junctions is the coaxial stacking of helices. This motif occurs when two separate helical elements stack to form coaxial helices as a pseudo-continuous helix. In addition, analysis from the solved crystal structures indicates that the RNA junctions can be classified into families according to their 3D shape or topology. The information obtained from coaxial stacking and topology (family) prediction can help predict RNA three-dimensional structures and gain a better understanding of RNA tertiary interactions.

By adopting methods and algorithms introduced in the previous chapter, a web server named Junction-Explorer is built. Given an RNA secondary structure in text format, the Junction-Explorer web server can identify and locate the junctions on the RNA secondary structure. For each identified RNA junction, the web server is able to predict the presence of helical coaxial stacking and the topology (family) of the junction. Junction-Explorer employs the random forests algorithm for prediction. The random forests classifier uses helical coaxial stacking and junction topology information from solved RNA 3D junctions as training data. Predictions are determined at the secondary structure level based on various features included in the classifier such as sequence, length, context, and thermodynamic parameters from RNA junctions. Junction-Explorer predicts coaxial stacks and topologies for both three and four-way junctions and only coaxial stacks for five-way and higher-order junctions. The Junction-Explorer web server with help document is freely accessible at http://bioinformatics.njit.edu/junction.

6.2 Method and Implementation

To predict coaxial stacking and junction family types (topologies) for three and four-way junctions, the features are extracted from a given RNA sequence and secondary structure such as the loop size within junctions, sequence content, and free-energy associated with base-stacking interactions between the base-pairs at the end of helices and their common loop region. Details of the features for these junctions are given in Chapter 5. Similar features are constructed for higher-order junctions. Specifically, H_i is used to represent the *i*-th helix according to the 5' to 3' orientation of the entire RNA secondary structure. The definition of a helix requires at least two consecutive Watson-Crick canonical base-pairs (G-C, A-U and G-U) to be formed. J_{ij} represents the loop region between helix H_i and helix H_j, and $|J_{ij}|$ denotes the number of nucleotides in (the size or length of) J_{ij}. If the $|J_{i(i+1)}|$ between a pair of neighboring helices H_i and H_{i+1} is small (e.g., < 5nt) then a coaxial stack is likely to exist between H_i and H_{i+1}; yet because a smaller loop size from neighboring loop regions can compete in the coaxial stacking formation, the minimum of the sizes of two neighboring loop regions are also included as a feature. Loop sizes are incorporated in ascending order to improve prediction accuracy. In addition, the maximum number of consecutive adenines for each loop region is included, as it has been reported that adenines in loops often form A-minor motifs on specific junction topologies.

The web server is implemented in C++, Perl-CGI, PHP, and R. The server accepts as input an RNA sequence along with its secondary structure whereby the secondary structure can be represented in bpseq format, CT format, or Vienna dot-bracket notation

[1,67]. The server identifies and locates the junctions in the input molecule. The feature values are then extracted from each identified junction. The server invokes the pre-trained classification program to determine the coaxial stacking and topology of each identified junction according to the junction's feature values. The classification program is implemented using the random forests package within R for statistical computing. Figure 6.1 is the flow chart of Junction-Explorer.

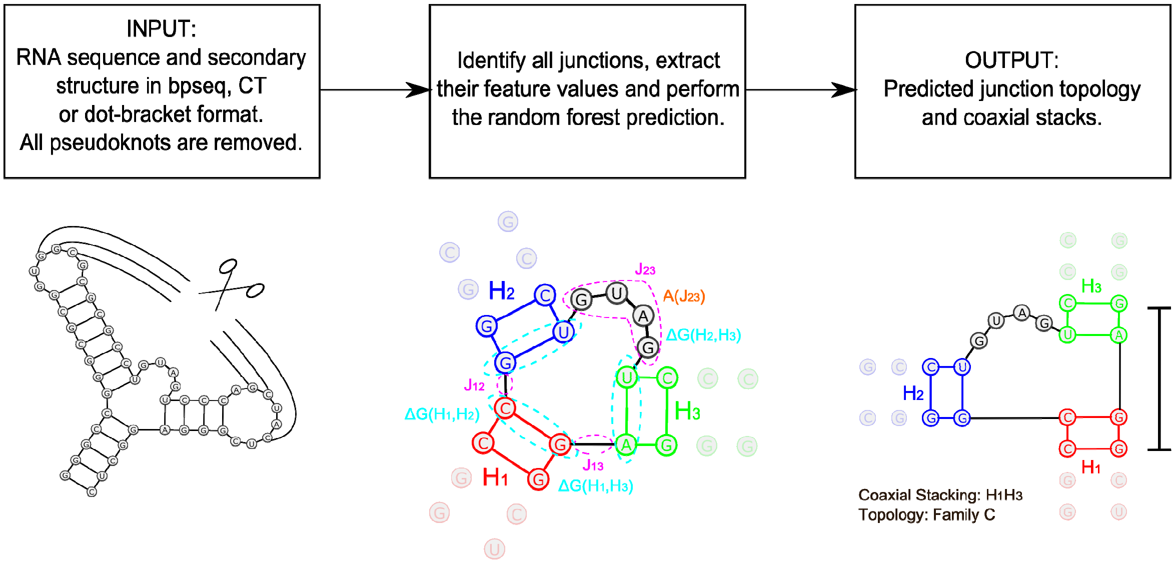
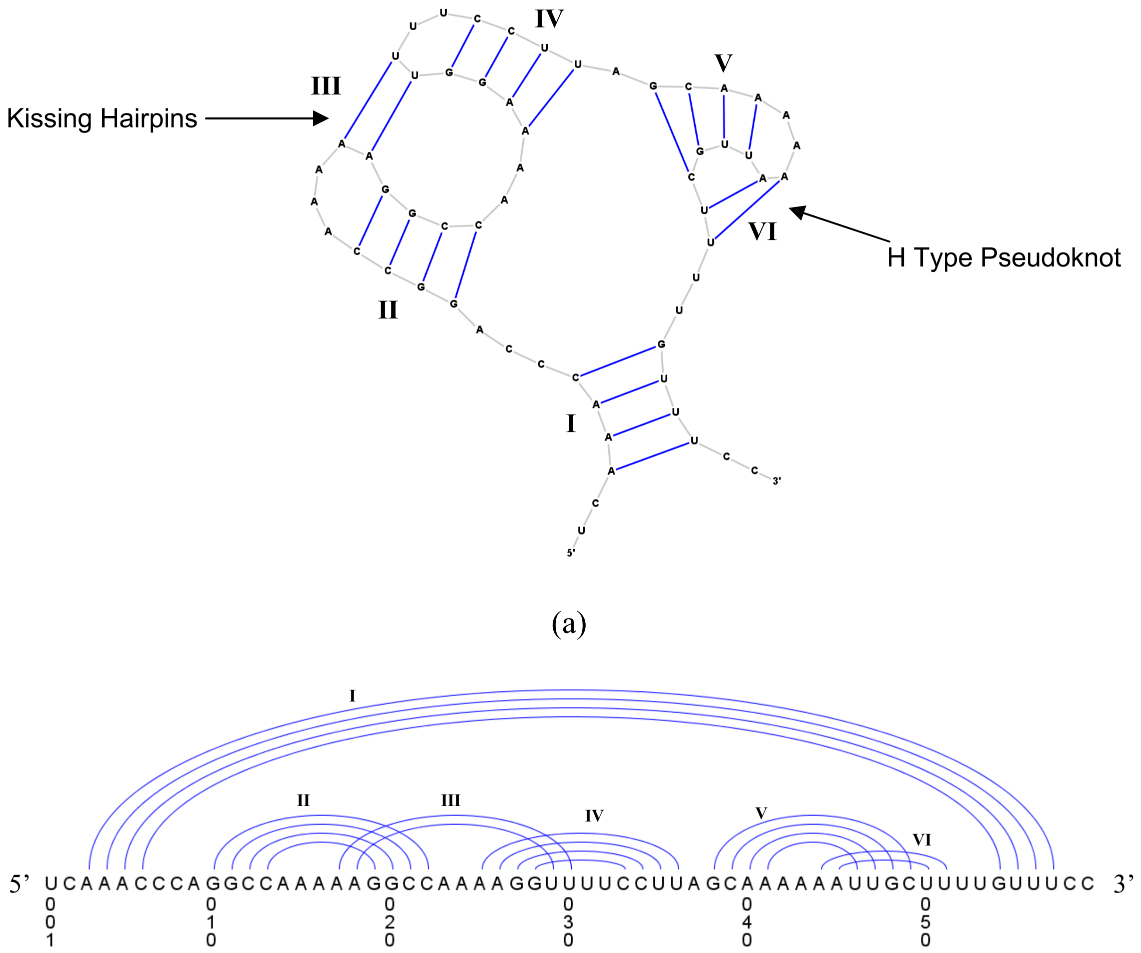

INPUT:
RNA sequence and secondary structure in bpseq, CT or dot-bracket format.
All pseudoknots are removed.
Identify all junctions, extract their feature values and perform the random forest prediction.
OUTPUT:
Predicted junction topology and coaxial stacks.
H2
J23
A(J23)
ΔG(H2,H3)
J12
ΔG(H1,H2)
H3
H1
J13
ΔG(H1,H3)
H3
H2
H1
Coaxial Stacking: H1H3
Topology: Family C

Figure 6.1 The flow chart of Junction-Explorer.

6.3 Pseudoknot Removal Algorithm

Since the pseudoknots may exist in the secondary structure which will cause the interference of junction identification, the web server uses K2N [68] for pseudoknot removal to make a pseudoknot-free secondary structure before performing the junction identification and the prediction. A simple pseudoknot removal algorithm is described below.

The definition of a pseudoknot on RNA secondary structures is a single base in any loop region (including hairpin loop, junction loop, internal loop and bulge loop) is paired with any base outside the loop region. That is, any two base pairs (i, j) and (i', j') form a pseudoknot when $i < i' < j < j'$. Figure 6.2 is an example of RNA secondary structure with two pseudoknots.



(a)

(b)

Figure 6.2 An example of RNA secondary structure with two pseudoknots (Kissing hairpins and H type Pseudoknot) in two different representations rendered by jViz.Rna 2.0 [69]. The six helices/stems are marked from I to VI. (a) The classical structure view. (b) The linear structure view. Pseudoknots form at the point where helices/stems are crossed.

In Figure 6.2b, pseudoknots form at the point where helices/stems are crossed. To remove the pseudoknot, some helices should be removed to avoid the crossing. Furthermore, in order to maintain the integrity of a structure, a helix with the smallest length should be removed from the crossing. Therefore, the goal of the pseudoknot removal algorithm works as follows:

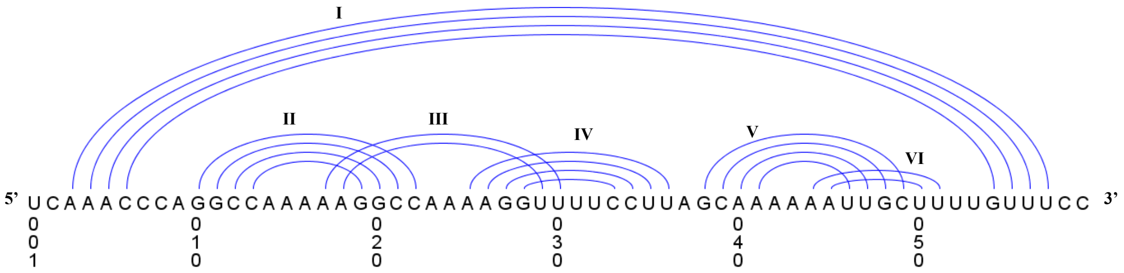
1. Calculate the score for each helix/stem. The score of a helix/stem *S* is calculated as the number of base pairs on this helix minus the total number of base pairs on helices that cross with *S*.
2. Remove the helix with the minimum score.
3. Go back to Step 1 until all pseudoknots are removed.

Figures 6.3, 6.4 and 6.5 show how the algorithm removes two pseudoknots from the RNA structure example shown in Figure 6.2.

6.4 Input and Output of Junction-Explorer

Junction-Explorer accepts as input an RNA sequence along with its secondary structure in one of the three formats: Bpseq format, CT format and Vienna dot-bracket format. The screenshot of input interface is presented in Figure 6.6a. The user takes the following three steps when using the web server:

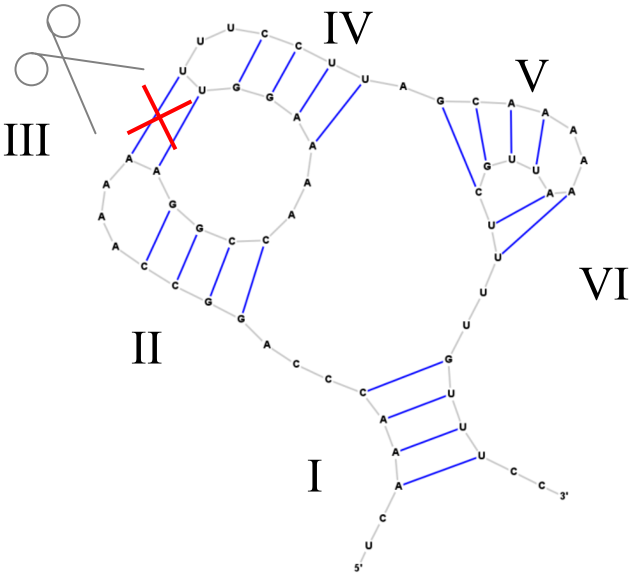
1. Paste an RNA sequence and its secondary structure represented in one of the three formats into the blank text field of the web server (or simply click any example button above the text field to retrieve an example RNA molecule).
2. Select the corresponding format option.
3. Click the “Submit” button.



(a)

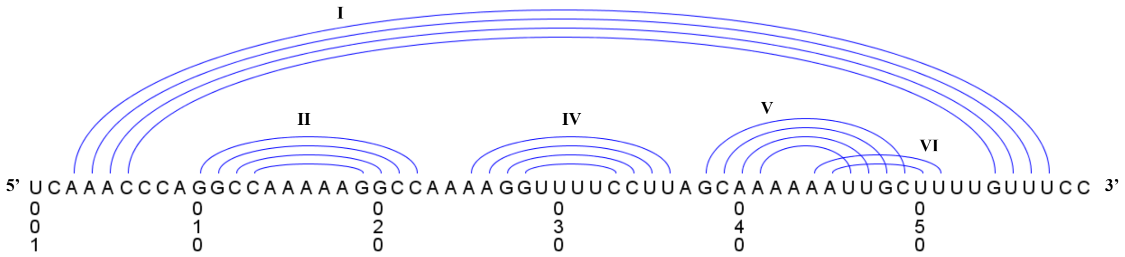
I:	4
II:	4 – 2 = 2
III:	2 – 4 – 4 = -6
IV:	4 – 2 = 2
V:	4 – 2 = 2
VI:	2 – 4 = -2

(b)



(c)

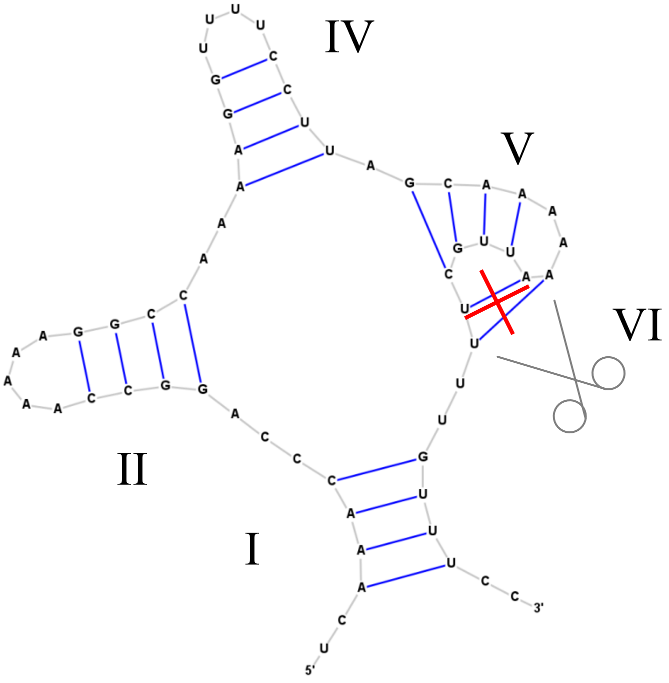
Figure 6.3 (a) The example of RNA secondary structure with two pseudoknots in linear structure view. (b) The score of each helix is calculated. The helix III is the one with the smallest score. (c) The helix III is removed.


I
II
IV
V
VI
5' UCAAACCCAGGCCAAAAAGGCCAAAAGGUUUCCUUAGCAAAAAAUUGCUUUGUUUCC 3'
001
010
020
030
040
050

(a)

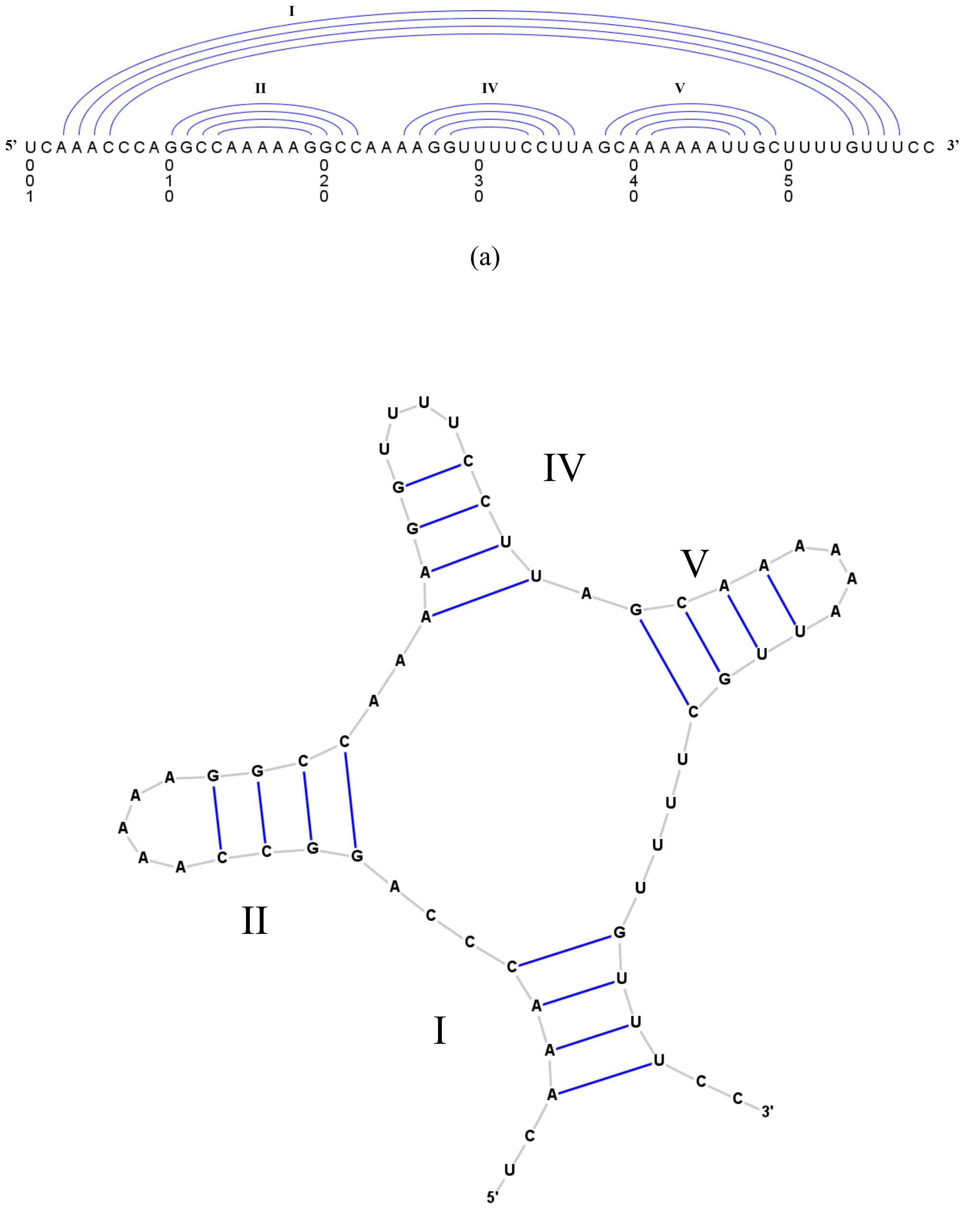
I:	4
II:	4
IV:	4
V:	4 – 2 = 2
VI:	2 – 4 = -2

(b)



(c)

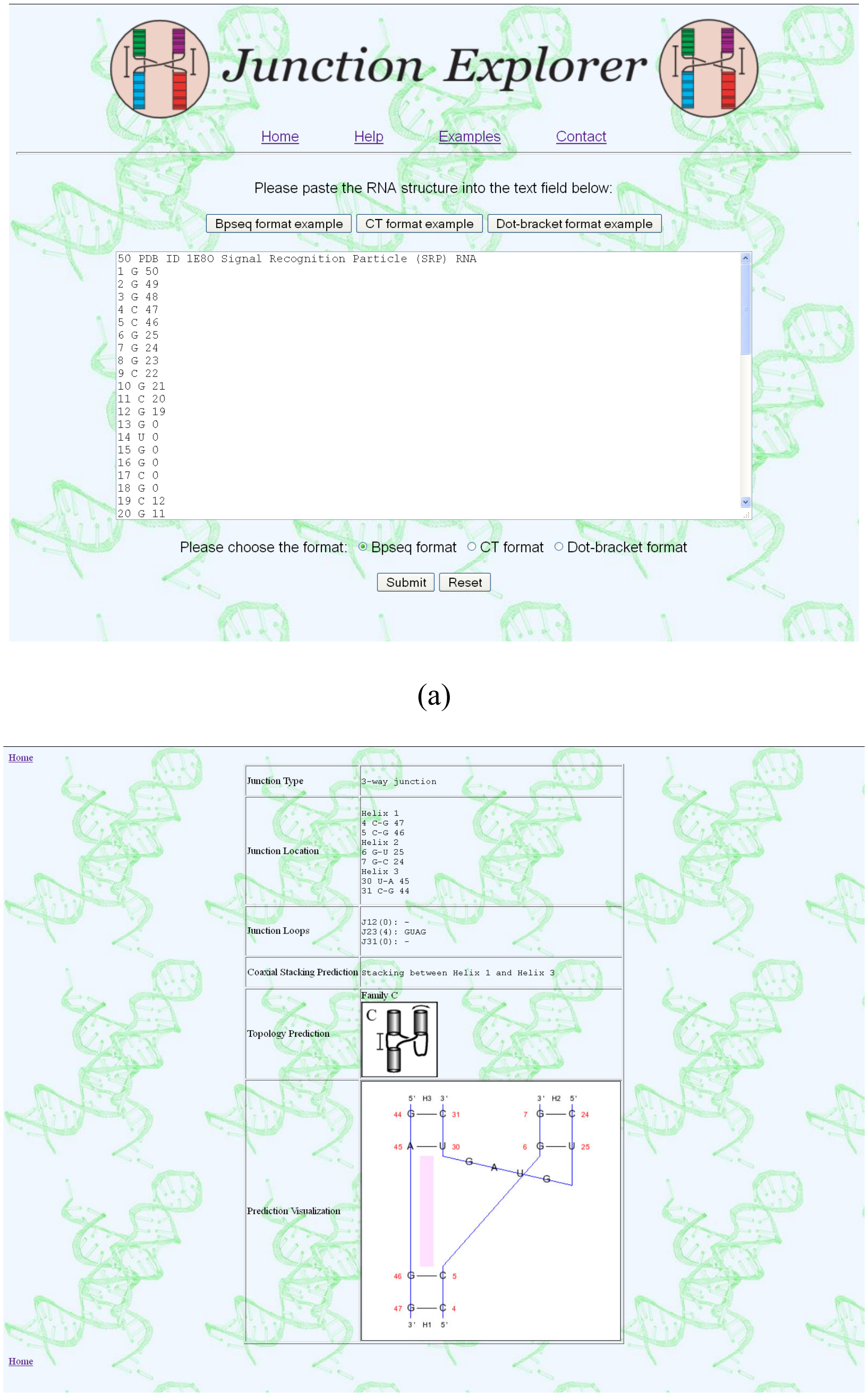
Figure 6.4 (a) This is the linear structure view after the helix III is removed. (b) Since there is still one pseudoknot existing, the score of each helix is calculated. The helix VI is the one with the smallest score. (c) The helix VI is removed.


I
II
IV
V
5' UCAAACCCAGGCCAAAAAGGCCAAAAGGUUUUCCUUAGCAAAAAAUUGCUUUGUUUCC 3'
001
010
020
030
040
050
IV
V
II
I
5'
3'

(a)

(b)

Figure 6.5 (a) This is the linear structure view after all pseudoknots are removed. (b) This is the classical structure view after all pseudoknots are removed.



Please paste the RNA structure into the text field below:

```
50 PDB ID 1E8O Signal Recognition Particle (SRP) RNA
1 G 50
2 G 49
3 G 48
4 C 47
5 C 46
6 G 25
7 G 24
8 G 23
9 C 22
10 G 21
11 C 20
12 G 19
13 G 0
14 U 0
15 G 0
16 G 0
17 C 0
18 G 0
19 C 12
20 G 11
```

Please choose the format: ● Bpseq format ○ CT format ○ Dot-bracket format

(a)

Junction Type	3-way junction
Junction Location	Helix 1 4 C-G 47 5 C-G 46 Helix 2 6 G-U 25 7 G-C 24 Helix 3 30 U-A 45 31 C-G 44
Junction Loops	J12(0): - J23(4): GUAG J31(0): -
Coaxial Stacking Prediction	Stacking between Helix 1 and Helix 3
Topology Prediction	Family C
Prediction Visualization	

(b)

Figure 6.6 (a) The screenshot of Junction-Explorer's input interface. (b) A screenshot of a Junction-Explorer's sample output.

After the user submits the RNA molecule, Junction-Explorer identifies and locates the junctions in the molecule and predicts the presence of coaxial helical stacking and the topology (family) of each junction in the input structure. The tool creates a detailed report, listing the type, location, loops, presence of predicted helical coaxial stacking, and predicted topology (family) of each identified junction in the molecule. A graphical display of predicted results for each junction is also presented, which allows the user to visualize the stack and family configuration in the junction. A screenshot of a sample output is presented in Figure 6.6b.

Usually, the web server displays the output on the web browser promptly. However, when the size of the input data is too large, processing the input structure becomes time-consuming. In this case, the web server provides a hyperlink instead. The user can access the predicted result through the hyperlink.

6.4.1 Input Format of Junction-Explorer

The user can input an RNA sequence and its secondary structure in one of the following three formats. If an RNA secondary structure has pseudoknots, it must be input in Bpseq or CT format, but not the Vienna dot-bracket format.

- Bpseq format: Here the first line constitutes the header of the format, listing the length and name of the input molecule. Subsequently multiple lines follow the header, wherein each line is comprised of three columns. The first column contains the position number of a nucleotide. This position number must start with 1. The second column contains the nucleotide name (A, C, G, or U). The third column contains the position number of the base with which the nucleotide is paired. If the nucleotide is not paired with any base, the third column is 0. A space must be used to separate the two neighboring columns.

- CT format: Here the first line constitutes the header of the format, which contains the length and name of the input molecule. In the CT format, each line consists of 6 columns. The first and sixth columns contain the position number of a nucleotide (base). This position number must start with 1. The third (fourth, respectively) column contains the position number minus one (plus one, respectively). The second column contains the nucleotide name (A, C, G, or U). The fifth column contains the position number of the base with which the nucleotide is paired. If the nucleotide is not paired with any base, the fifth column is 0. A tab must be used to separate the two neighboring columns.
- Vienna dot-bracket format: Here the first line constitutes the header, which starts with the ">" character followed by the name of the input molecule. The second line contains the input sequence from 5' to 3'. The third line contains the secondary structure of the input sequence, where a base pair is represented by an opening and closing bracket and an unpaired base is represented by a dot.

6.4.2 Output Format of Junction-Explorer

The web server displays a table for each identified junction, listing the following information concerning the junction. If no junction is identified, the web server displays a message so indicating. An example of the predicted results is shown in Figure 6.6b.

- Junction Type: This field shows the type of the junction, which can be three-way, four-way, five-way or of a higher-order, depending on how many helices are involved.

- Junction Location: This field shows the nucleotides and their positions for the helices involved in the junction. These positions define the location of the junction. For each helix, only the two consecutive base pairs that are closest to the junction are displayed.
- Junction Loops: This field shows the size and the nucleotides in the loop region between every two neighboring helices. For example, “J23 (4): GUAG” means that the loop region between Helix 2 and Helix 3 contains four nucleotides G, U, A and G. As another example, “J12 (0): -” indicates that the loop region between Helix 1 and Helix 2 has zero nucleotide.
- Coaxial Stacking Prediction: This field shows the predicted outcome for coaxial stacking.
- Topology Prediction: This field shows the predicted outcome for junction family.
- Prediction Visualization: This field presents a graphical display of the predicted coaxial stacking of helices and predicted topology for the junction.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Contributions and Conclusions

In addition to the traditional role of RNA sequential motifs, RNA secondary or tertiary structure motifs play important roles in cells. However, until today, very few online web servers were available for RNA motif search and prediction. In this dissertation, a cyberinfrastructure named RNAcyber is proposed, designed and implemented, which is capable of performing RNA motif search and prediction. The RNAcyber infrastructure is fully operational, with all of its components accessible on the Internet.

In Chapter 2, the first component of RNAcyber is introduced, which is a web-based search engine named RmotifDB. This web-based tool integrates an RNA secondary structure comparison algorithm with the secondary structure motifs stored in the Rfam database. With a user-friendly interface, RmotifDB provides the ability to search for ncRNA structure motifs in both structural and sequential ways. The second component of RNAcyber is an enhanced version of RmotifDB, which is introduced in Chapter 3. This enhanced version combines data from multiple sources, incorporates a variety of well-established structure-based search methods, and is integrated with the Gene Ontology. To display RmotifDB's search results, a software tool, called RSview, is developed. RSview is able to display the search results in a graphical manner, which is described in Chapter 4.

The important application of secondary structure motif search includes finding ncRNA motifs similar to newly discovered motifs from ncRNA gene in a fast way; especially when motifs are related with biological functions or diseases. Furthermore, by motifs searching over the databases, scientists may discover and explore more motif-relevant information such as RNA type, gene id, species name, gene segment location and gene ontology, which may never been explored before.

In Table 7.1, it shows the function comparison between RmotifDB web server and closely relevant programs/servers such as Rfam [4], RSmatch [9], RADAR [14] and UTRdb [48]. In this table, several different functions with varied aspects are examined and compared. All tools, except RSmatch, contain a motif database and a convenient web interface, but only RmotifDB's database was integrated with the Gene Ontology information. RmotifDB, RSmach and RADAR are able to perform a secondary structure search, but Rfam and UTRdb can only perform a sequential search. Lastly, only RmotifDB allows its users to submit new data through a web submission system. Through this table, it proves that RmotifDB is strongly comparable with other closely related tools.

Table 7.1 The Function Comparison between RmotifDB Web Server and Closely Relevant Programs/Servers

	RmotifDB	**Rfam[4]**	**RSmatch[9]**	**RADAR[14]**	**UTRdb[48]**
Motif Database	Yes	Yes	No	Yes	Yes
Gene Ontology	Yes	No	No	No	No
Secondary Structure Search	Yes	No	Yes	Yes	No
Web Interface	Yes	Yes	No	Yes	Yes
New Data Submission	Yes	No	No	No	No

Finally, in Chapters 5 and 6, RNAcyber contains a web-based tool called Junction-Explorer, which employs a data mining method for predicting tertiary motifs in RNA junctions. Specifically, the tool is trained on solved RNA tertiary structures obtained from the Protein Data Bank, and is able to predict the configuration of coaxial helical stacks and families (topologies) in RNA junctions at the secondary structure level. Junction-Explorer employs several algorithms for motif prediction, including a random forest classification algorithm, a pseudoknot removal algorithm, and a feature ranking algorithm based on the gini impurity measure. A series of experiments including 10-fold cross-validation were conducted to evaluate the performance of the Junction-Explorer tool. Experimental results demonstrate the effectiveness of the proposed algorithms and the superiority of the tool over existing methods. While data analysis results were reported previously, to the best of current knowledge, this is the first web server capable of performing the predictions online. The server provides an important step toward RNA 3D structure modeling and understanding. Predictions made by the web server can add reasonable constraints to the conformational space of RNA three-dimensional structures.

7.2 Future Work

In future work, the plan is to develop new data mining and data integration techniques for finding RNA structural motifs in various organisms and for integrating the motifs with several biomedical ontologies beyond Gene Ontology. Advanced search methods that combine statistical methods with efficient data structures or algorithms for a high-recall and high-precision search engine for RNA tertiary motifs are under development. Another future work includes the development of new methods for predicting other RNA tertiary motifs such as A-minors, pseudoknots or ribose zippers and their interactions with junctions. Furthermore, the higher-order junctions, as well as proteins' interactions within junctions, have not been fully explored.

REFERENCES

1. Hofacker,I.L. (2003) Vienna RNA secondary structure server. *Nucleic Acids Res.*, **31**, 3429-3431.

2. Marzluff,W.F. and Duronio,R.J. (2002) Histone mRNA expression: multiple levels of cell cycle regulation and important developmental consequences. *Curr. Opin. Cell Biol.*, **14**, 692-699.

3. Pesole,G., Liuni,S., Grillo,G., Licciulli,F., Mignone,F., Gissi,C. and Saccone,C. (2002) UTRdb and UTRsite: specialized databases of sequences and functional elements of 5' and 3' untranslated regions of eukaryotic mRNAs. *Nucleic Acids Res.*, **30**, 335-340.

4. Jones,S.G., Moxon,S., Marshall,M., Khanna,A., Eddy,S.R. and Bateman,A. (2005) Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res.*, **33**, D121-D124.

5. Andronescu,M., Bereg,V., Hoos,H.H. and Condon,A. (2008) RNA STRAND: The RNA Secondary Structure and Statistical Analysis Database. *BMC Bioinformatics*, **9**:340.

6. Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, **85**, 2444–2448.

7. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403-410.

8. Crooks,G.E., Hon,G., Chandonia,J.M. and Brenner,S.E. (2004) WebLogo: A sequence logo generator. *Genome Res.*, **14**, 1188-1190.

9. Liu,J., Wang,J.T.L., Hu,J. and Tian,B. (2005) A method for aligning RNA secondary structures and its application to RNA motif detection. *BMC Bioinformatics*, **6**:89.

10. Klein,R.J. and Eddy,S.R. (2003) RSEARCH: Finding Homologs of Single Structured RNA Sequences. *BMC Bioinformatics*, **4**:44.

11. Höchsmann,M., Voss,B. and Giegerich,R. (2004) Pure Multiple RNA Secondary Structure Alignments: A Progressive Profile Approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **1**, 53-62.

12. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235-242.

13. Eddy,S.R. (2002) A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics*, **3**:18.

14. Khaladkar,M., Bellofatto,V., Wang,J.T.L., Tian,B. and Zhang,K. (2006) RADAR: an interactive web-based toolkit for RNA data analysis and research. *Proc. of the 6th IEEE Symposium on Bioinformatics and Bioengineering*, 209-212.

15. Wilkie,G.S., Dickson,K.S. and Gray,N.K. (2003) Regulation of mRNA translation by 5'- and 3'-UTR-binding factors. *Trends Biochem. Sci.*, **28**, 182-188.

16. Bakheet,T., Frevel,M., Williams,B.R., Greer,W. and Khabar,K.S. (2001) ARED: human AU-rich element-containing mRNA database reveals an unexpectedly diverse functional repertoire of encoded proteins. *Nucleic Acids Res.*, **29**, 246-254.

17. Lewis,B.P., Shih,I.H., Jones-Rhoades,M.W., Bartel,D.P. and Burge,C.B. (2003) Prediction of mammalian microRNA targets. *Cell*, **115**, 787-798.

18. Wang,J.T.L., Shapiro,B.A. and Shasha,D. (eds.) (1999) *Pattern Discovery in Biomolecular Data: Tools, Techniques and Applications*. Oxford University Press, New York.

19. Wang,J.T.L., Wu,C.H. and Wang,P.P. (eds.) (2003) *Computational Biology and Genome Informatics.* World Scientific Publishing Company, Singapore.

20. Wang,J.T.L., Zaki,M.J., Toivonen,H.T.T. and Shasha,D. (eds.) (2005) *Data Mining in Bioinformatics*. Springer-Verlag, London.

21. Akmaev,V.R., Kelley,S.T. and Stormo,G.D. (2000) Phylogenetically enhanced statistical tools for RNA structure prediction. *Bioinformatics*, **16**, 501-512.

22. Bindewald,E. and Shapiro,B.A. (2006) RNA secondary structure prediction from sequence alignments using a network of k-nearest neighbor classifiers. *RNA*, **12**, 342-352.

23. Bindewald,E., Schneider,T.D. and Shapiro,B.A. (2006) CorreLogo: an online server for 3D sequence logos of RNA and DNA alignments. *Nucleic Acids Res.*, **34**, 405-411.

24. Gorodkin,J., Stricklin,S.L. and Stormo,G.D. (2001) Discovering common stem-loop motifs in unaligned RNA sequences. *Nucleic Acids Res.*, **29**, 2135-2144.

25. Blake,J.A. and Bult,C.J. (2006) Beyond the data deluge: data integration and bio-ontologies. *Journal of Biomedical Informatics*, **39**, 314-320.

26. Coulet,A., Smail-Tabbone,M., Benlian,P., Napoli,A. and Devignes,M.-D. (2006) SNP-Converter: an ontology-based solution to reconcile heterogeneous SNP descriptions for pharmacogenomic studies. *Proc. of the 3rd International Workshop on Data Integration in the Life Sciences*, Hinxton, UK, 82-93.

27. Gupta,A. and Santini,S. (2006) On querying OBO ontologies using a DAG pattern query language. *Proc. of the 3rd International Workshop on Data Integration in the Life Sciences*, Hinxton, UK, 152-167.

28. Mork,P., Shaker,R. and Tarczy-Hornoch,P. (2005) The multiple roles of ontologies in the biomediator data integration system. *Proc. of the 2nd International Workshop on Data Integration in the Life Sciences*, San Diego, USA, 96-104.

29. Tong,R., Quackenbush,J. and Snuffin,M. (2003) Knowledge-based access to the bio-medical literature, ontologically-grounded experiments for the TREC 2003 genomics track. *Proc. of the 12th Text Retrieval Conference*, 547-551.

30. Chang,C.-Y., Wang,J.T.L. and Chang,R.K. (1998) Scientific data mining: a case study. *International Journal of Software Engineering and Knowledge Engineering*, **8**, 77-96.

31. Wang,J.T.L., Shapiro,B.A., Shasha,D., Zhang,K. and Chang,C.-Y. (1996) Automated discovery of active motifs in multiple RNA secondary structures. *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining*, 70-75.

32. Wang,J.T.L., Shapiro,B.A., Shasha,D., Zhang,K. and Currey,K.M. (1998) An algorithm for finding the largest approximately common substructures of two trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 889-895.

33. Wang,J.T.L., Rozen,S., Shapiro,B.A., Shasha,D., Wang,Z. and Yin,M. (1999) New techniques for DNA sequence classification. *Journal of Computational Biology*, **6**, 209-218.

34. Grillo,G., Licciulli,F., Liuni,S., Sbisa,E. and Pesole,G. (2003) PatSearch: a program for the detection of patterns and structural motifs in nucleotide sequences. *Nucleic Acids Res.*, **31**, 3608-3612.

35. Pruitt,K.D., Katz,K.S., Sicotte,H. and Maglott,D.R. (2000) Introducing RefSeq and LocusLink: curated human genome resources at the NCBI. *Trends Genet.*, **16**, 44-47.

36. Dalgaard,P. (2004) *Introductory Statistics with R*. Springer, New York.

37. Cohen-Boulakia,S., Davidson,S.B. and Froidevaux,C. (2005) A user-centric framework for accessing biological sources and tools. *Proc. of the 2nd International Workshop on Data Integration in the Life Sciences*, San Diego, USA, 3-18.

38. Davidson,S.B., Crabtree,J., Brunk,B.P., Schug,J., Tannen,V., Overton,G.C. and Stoeckert,C.J.,Jr. (2001) K2/Kleisli and GUS: experiments in integrated access to genomic data sources. *IBM Systems Journal*, **40**, 512-531.

39. Eckman,B.A., Kosky,A. and Laroco,L.A.,Jr. (2001) Extending traditional query-based integration approaches for functional characterization of post-genomic data. *Bioinformatics*, **17**, 587-601.

40. Hunt,E., Pafilis,E., Tulloch,I. and Wilson,J. (2004) Index-driven XML data integration to support functional genomics. *Proc. of the 1st International Workshop on Data Integration in the Life Sciences*, Leipzig, Germany, 95-109.

41. Huttenhower,C., Hibbs,M., Myers,C. and Troyanskaya,O.G. (2006) A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, **22**, 2890-2897.

42. Krishnamurthy,L., Nadeau,J.H., Ozsoyoglu,G., Ozsoyoglu,Z.M., Schaeffer,G., Tasan,M. and Xu,W. (2003) Pathways database system: an integrated system for biological pathways. *Bioinformatics*, **19**, 930-937.

43. Lacroix,Z., Raschid,L. and Eckman,B.A. (2004) Techniques for optimization of queries on integrated biological resources. *J. Bioinformatics and Computational Biology*, **2**, 375-412.

44. Leser,U., Lehrach,H. and Crollius,H.R. (1998) Issues in developing integrated genomic databases and application to the human X chromosome. *Bioinformatics*, **14**, 583-590.

45. Markowitz,V.M. (2006) An application driven perspective on biological data integration. *Proc. of the 3rd International Workshop on Data Integration in the Life Sciences*, Hinxton, UK, 1.

46. McPhillips,T.M., Bowers,S. and Ludaescher,B. (2006) Collection-oriented scientific workflows for integrating and analyzing biological data. *Proc. of the 3rd International Workshop on Data Integration in the Life Sciences*, Hinxton, UK, 248-263.

47. Wang,J.T.L. and Wu,X. (2006) Kernel design for RNA classification using support vector machines. *International Journal of Data Mining and Bioinformatics*, **1**, 57-76.

48. Mignone,F., Grillo,G., Licciulli,F., Iacono,M., Liuni,S., Kersey,P.J., Duarte,J., Saccone,C. and Pesole,G. (2005) UTRdb and UTRsite: a collection of sequences and regulatory motifs of the untranslated regions of eukaryotic mRNAs. *Nucleic Acids Res.*, **33**, D141-D146.

49. Yang,H., Jossinet,F., Leontis,N., Chen,L., Westbrook,J., Berman,H.M. and Westhof,E. (2003) Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Research*, **31**, 3450-3460.

50. Xin,Y., Laing,C., Leontis,N.B. and Schlick,T. (2008) Annotation of tertiary interactions in RNA structures reveals variations and correlations. *RNA*, **14**, 2465-2477.

51. Leontis,N.B. and Westhof,E. (2001) Geometric nomenclature and classification of RNA base pairs. *RNA*, **7**, 499-512.

52. Laing,C. and Schlick,T. (2010) Computational approaches to 3D modeling of RNA. *J. Phys.: Condens. Matter*, **22**, 283101.

53. Bindewald,E., Hayes,R., Yingling,Y.G., Kasprzak,W. and Shapiro,B.A. (2008) RNAJunction: a database of RNA junctions and kissing loops for three-dimensional structural analysis and nanodesign. *Nucleic Acids Res.*, **36**, D392-D397.

54. Ouellet,J., Melcher,S., Iqbal,A., Ding,Y. and Lilley,D.M.J. (2010) Structure of the three-way helical junction of the hepatitis C virus IRES element. *RNA*, **16**, 1597-1609.

55. Kim,S.H., Sussman,J.L., Suddath,F.L., Quigley,G.J., McPherson,A., Wang,A.H., Seeman,N.C. and Rich,A. (1974) The general structure of transfer RNA molecules. *Proc. Natl. Acad. Sci. USA*, **71**, 4970-4974.

56. Laing,C. and Schlick,T. (2009) Analysis of four-way junctions in RNA structures. *J. Mol. Biol.*, **390**, 547-559.

57. Laing,C., Jung,S., Iqbal,A. and Schlick,T. (2009) Tertiary motifs revealed in analyses of higher-order RNA junctions. *J. Mol. Biol.*, **393**, 67-82.

58. Lescoute,A. and Westhof,E. (2006) Topology of three-way junctions in folded RNAs. *RNA*, **12**, 83-93.

59. Jossinet,F. and Westhof,E. (2005) Sequence to Structure (S2S): display, manipulate and interconnect RNA data from sequence to structure. *Bioinformatics*, **21**, 3320-3321.

60. Nissen,P., Ippolito,J.A., Ban,N., Moore,P.B. and Steitz,T.A. (2001) RNA tertiary interactions in the large ribosomal subunit: the A-minor motif. *Proc. Natl. Acad. Sci. USA*, **98**, 4899−4903.

61. Reuter,J.S. and Mathews,D.H. (2010). RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, **11**:129.

62. Tyagi,R. and Mathews,D.H. (2007) Predicting helical coaxial stacking in RNA multibranch loops. *RNA*, **13**, 939-951.

63. Mathews,D.H., Disney,M.D., Childs,J.L., Schroeder,S.J., Zuker,M. and Turner,D.H. (2004) Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. USA*, **101**, 7287−7292.

64. Mathews,D.H., Sabina,J., Zuker,M. and Turner,D.H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911−940.

65. Breiman,L. (2001) Random forests. *Machine Learning*, **45**, 5-32.

66. Liaw,A. and Wiener,M. (2002) Classification and regression by random forest. *R News*, **2**, 18-22.

67. Cannone,J.J., Subramanian,S., Schnare,M.N., Collett,J.R., D'Souza,L.M., Du,Y., Feng,B., Lin,N., Madabusi,L.V., Müller,K.M., Pande,N., Shang,Z., Yu,N. and Gutell,R.R. (2002) The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics*, **3**:2.

68. Smit,S., Rother,K., Heringa,J. and Knight,R. (2008) From knotted to nested RNA structures: a variety of computational methods for pseudoknot removal. *RNA*, **14**, 410-416.

69. Wiese,K.C., Glen,E. and Vasudevan,A. (2005) jViz.Rna - a Java tool for RNA secondary structure visualization. *IEEE Transactions on NanoBioscience*, **4**, 212-218.