Classification of RNA Pseudoknots and Comparison of Structure Prediction Methods

Cong Zeng

Université Paris-Sud
École Doctorale : Informatique
Laboratoire : Laboratoire de Recherche en Informatique (LRI)

Discipline : Bioinformatique

Thèse de doctorat

ZENG Cong

Classification of RNA Pseudoknots and Comparison of Structure Prediction Methods

Date de soutenance : 03/07/2015

Directeur de thèse:

M. DENISE Alain	PR (LRI, Université Paris-Sud)

Composition du jury:

Rapporteurs :	M. BLIN Guillaume	PR (LaBRI, Université Bordeaux 1)
	Mme. GASPIN Christine	DR (MIAT, INRA Toulouse)
Examinateurs :	M. SABOURET Nicolas	PR (LIMSI, Université Paris-Sud)
	Mme. TAHI Fariza	MCF (IBISC, Université d'Evry-Val d'Essonne)
	M. NAMY Olivier	CR1 (I2BC, CNRS and Université Paris-Sud)

Abstract

Lots of researches convey the importance of the RNA molecules, as they play vital roles in many molecular procedures. And it is commonly believed that the structures of the RNA molecules hold the key to the discovery of their functions.

During the investigation of RNA structures, the researchers are dependent on the bioinformatical methods increasingly. Many *in silico* methods of predicting RNA secondary structures have emerged in this big wave, including some ones which are capable of predicting pseudoknots, a particular type of RNA secondary structures.

The purpose of this dissertation is to try to compare the state-of-the-art methods predicting pseudoknots, and offer the colleagues some insights into how to choose a practical method for the given single sequence. In fact, lots of efforts have been done into the prediction of RNA secondary structures including pseudoknots during the last decades, contributing to many programs in this field. Some challenging questions are raised consequently. How about the performance of each method, especially on a particular class of RNA sequences? What are their advantages and disadvantages? What can we benefit from the contemporary methods if we want to develop new ones? This dissertation holds the confidence in the investigation of the answers.

This dissertation carries out quite many comparisons of the performance of predicting RNA pseudoknots by the available methods. One main part focuses on the prediction of frameshifting signals by two methods principally. The second main part focuses on the prediction of pseudoknots which participate in much more general molecular activities.

In detail, the second part of work includes 414 pseudoknots, from both the Pseudobase and the Protein Data Bank, and 15 methods including 3 exact meth-

ods and 12 heuristic ones. Specifically, three main categories of complexity measurements are introduced, which further divide the 414 pseudoknots into a series of subclasses respectively.

The comparisons are carried out by comparing the predictions of each method based on the entire 414 pseudoknots, and the subsets which are classified by both the complexity measurements and the length, RNA type and organism of the pseudoknots.

The result shows that the pseudoknots in nature hold a relatively low complexity in all measurements. And the performance of contemporary methods varies from subclass to subclass, but decreases consistently as the complexity of pseudoknots increases. More generally, the heuristic methods globally outperform the exact ones. And the susceptible assessment results are dependent strongly on the quality of the reference structures and the evaluation system. Last but not least, this part of work is provided as an on-line benchmark for the bioinformatics community.

Résumé

De nombreuses recherches ont constaté l'importance des molécules d'ARN, car ils jouent un rôle vital dans beaucoup de procédures moléculaires. Et il est accepté généralement que les structures des molécules d'ARN sont la clé de la découverte de leurs fonctions.

Au cours de l'enquête de structures d'ARN, les chercheurs dépendent des méthodes bioinformatiques de plus en plus. Beaucoup de méthodes *in silico* de prédiction des structures secondaires d'ARN ont émergé dans cette grosse vague, y compris certains qui sont capables de prédire pseudo-nœuds, un type particulier de structures secondaires d'ARN.

Le but de ce travail est d'essayer de comparer les méthodes de l'état de l'art pour prédiction de pseudo-nœud, et offrir aux collègues des idées sur le choix d'une méthode pratique pour la seule séquence donnée. En fait, beaucoup d'efforts ont été fait dans la prédiction des structures secondaires d'ARN parmi lesquelles le pseudo-nœud les dernières décennies, contribuant à de nombreux programmes dans ce domaine. Certaines enjeux sont soulevées conséquemment. Comment est-elle la performance de chaque méthode, en particulier sur une classe de séquences d'ARN particulière? Quels sont leurs pour et contre? Que pout-on profiter des méthodes contemporaines si on veut développer de nouvelles? Cette thèse a la confiance dans l'enquÃłte sur les réponses.

Cette thèse porte sur très nombreuses comparaisons de la performance de prédire pseudo-nœuds d'ARN par les méthodes disponibles. Une partie principale se concentre sur la prédiction de signaux de déphasage par deux méthodes principalement. La deuxième partie principale se concentre sur la prédiction de pseudo-nœuds qui participent à des activités moléculaires beaucoup plus générale.

Dans le détail, la deuxième partie du travail comprend 414 pseudo-nœuds de

Pseudobase et de la Protein Data Bank, ainsi que 15 méthodes dont 3 méthodes exactes et 12 heuristiques. Plus précisément, trois grandes catégories de mesures complexes sont introduites, qui divisent encore les 414 pseudo-nœuds en une série de sous-classes respectivement.

Les comparaisons se passent par comparer les prédictions de chaque méthode basée sur l'ensemble des 414 pseudo-nœuds, et les sous-ensembles qui sont classés par les deux mesures complexes et la longueur, le type de l'ARN et de l'organisme des pseudo-nœuds.

Le résultat montre que les pseudo-nœuds portent une complexité relativement faible dans toutes les mesures. Et la performance des méthodes modernes varie de sous-classe à l'autre, mais diminue constamment lors que la complexité de pseudo-nœuds augmente. Plus généralement, les méthodes heuristiques sont supérieurs globalement à celles exacts. Et les résultats de l'évaluation sensibles sont dépendants fortement de la qualité de structure de référence et le système d'évaluation. Enfin, cette partie du travail est fourni comme une référence en ligne pour la communauté bioinformatique.

Acknowledgments

I am deeply grateful to my supervisor DENISE Alain, who was excellent mentor, advisor, supporter and friend throughout my graduate studies. I appreciate his kindness, patience, and encouragements he gave to me, from which I will benefit all my life.

I thank my reviewers BLIN Guillaume and GASPIN Christine, and the other committee members SABOURET Nicolas, TAHI Fariza, and NAMY Olivier for their attendance of my defence, and invaluable comments and suggestions on my work.

I thank our bioinformatics group, Christine and all the colleagues for their discussions, collaborations and friendship.

I also would like to thank many other colleagues in the community for their kindness and explanation of my questions, which have helped make this work possible.

Thanks to the China Scholarship Council, who provided financial support during my graduate studies.

Finally, many thanks to my family for their encouragements and confidence. And thanks to my dear friends for supporting and accompanying, although I do not list all your names, and Ming for his love.

This dissertation is dedicated to my beloved grandparents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation focuses on the identification of pseudoknots, a secondary structural motif of RNA, including principally the study of the hierarchical classifications of pseudoknots, and the comparison of mechanisms and performances of the methods that are available to predict pseudoknots.

Pseudoknots are involved in a variety of molecular processes, such as playing the role as a stimulator in the programmed ribosomal frameshifting, one classical recoding event where the ribosome can switch to an alternative open reading frame such that a different peptide is translated.

The repertoire of pseudoknots includes the participation in more general molecular activities. These versatile motifs are publicly accessible via the *PseudoBase* [Van Batenburg et al., 2000], a particular database for pseudoknots, and the *Protein Data Bank (PDB)* [Berman et al., 2000], a database with some entries containing pseudoknots. The pseudoknots from both provenances are well anatomized in this dissertation, including their classification in accordance to several measurements of complexity hierarchically.

Based on the entries from the two databases, a series of comparisons are carried out to verify the flexibility of the contemporary methods in predicting the pseudoknots. The evaluation of performance of prediction will be performed from the perspectives of the characteristics of both the pseudoknots and the prediction methods. In other words, the performance of prediction is revealed from the evaluation of sub-collections of predictions which are separated in accordance to the length of the sequences, the complexities of the pseudoknots, the mechanism of

the prediction methods etc.

In detail, the dissertation is organized as follows.

Chapter 2 is a brief introduction about the background of RNAs and two types of RNA secondary structures, the standard pseudoknot-free secondary structures and the pseudoknots. Couples of RNA file formats employed in bioinformatics are introduced as well, which encompass both the sequential and structural information of the given RNA. The pseudoknot pattern and the linear graphical representation of the pseudoknots are introduced in this chapter, which two serve as the principal demonstration of the pseudoknots in the following chapters.

Chapter 3 is a general description of the state-of-the-art researches on the prediction of RNA secondary structures. Predicting RNA secondary structures can take advantage of the comparative methods, with the assistance of sequence or structural alignment, but this dissertation focuses on the methods predicting RNA secondary structure from a single given sequence. The methods may employ the mechanisms of minimizing the free energy of the RNA folding, maximizing the number of base pairs, calculating the partition functions and probability of base pairs, or some heuristic strategies to detect a best secondary structure in the defined model. However it has been proved that predicting an RNA secondary structure containing arbitrary pseudoknots is NP-hard. Consequently, each method that can perform the prediction in polynomial time has different levels of compromise between the computational cost and agreeable performance, such as the pseudoknot types that are recognized. On the other hand, heuristic methods may remedy the restriction on the types and the lengths of pseudoknots that can be detected by the particular searching model, but with a sacrifice on the optimality of the detection.

Chapter 4 describes a cooperative work of detecting the -1 programmed ribosomal frameshifting (-1 PRF) signals. The ribosomal frameshiftings are one classical recoding event occurring in the regulation of post transcription. A frameshifting signal contains two primary components, the slippery sequence and the downstream secondary structure as a stimulator. The pseudoknot is declared to promote a frameshifting more efficiently than the standard stem-loop secondary structure, especially in the viruses [Brierley, 1995; Brierley et al., 2007]. Several

algorithms detecting the frameshifting signals, and the comparison of their performance of prediction are introduced. Particularly, as a significant part of the comparisons, Orphea [Brégeon et al.; Forest, 2005], a software developed by the LRI and IGM groups, and KnotInFrame [Theis et al., 2008], a pipeline for detecting the frameshifting signals by a German group, the two programs are compared for their detection of frameshifting signals based on 34 frameshifting signals in Pseudobase.

The pseudoknots involved in the frameshifting recoding events are a subset of the pseudoknot family. Chapters 5, 6 and 7 illustrates the study of much more general pseudoknots and the prediction methods. The study is carried out in two main categories. First, a set of classifications of the RNA pseudoknots are introduced, covering the physical interactions, the algorithmic accessibilities and the conformational characteristics. Then, a benchmark of predicting pseudoknots by the state-of-the-art methods is shown. The predictions are evaluated with the criteria of the *sensitivity*, the *positive predictive value (PPV)* and the *Matthews correlation coefficient (MCC)*, in a variety of the separated sub-collections of pseudoknots which are divided with respect to the lengths and complexities of the pseudoknots, the RNA families they belong to etc. Additionally, the general performance of the exact methods and the heuristic methods, will be compared as well. The benchmark can be expected as a database of the knowledges which are learned from the anatomization of the pseudoknots in hand and the predictions based on them by the state-of-the-art capable methods. The knowledges or experiences obtained cater to the motivation of the benchmark, which is at the service of giving a hand to the users who are interested in the prediction of RNA pseudoknots *in silico*, helping or guiding them to accomplish the mission of *how to predict a plausible pseudoknot from the given RNA sequence?*

In practice, Chapter 5 introduces the preparation part of this benchmark. It covers the motivation of this study, the dataset of pseudoknots, three pseudoknot complexity measures which classify the dataset hierarchically, the prediction methods, and the evaluation parameters. Chapter 6 shows the results of this benchmark. It includes both the classification of pseudoknots and the prediction of pseudoknots by the considered methods. And the web development of

the benchmark is introduced next, which allows the results being available to the community on-line. Chapter 7 discusses the results, and concludes the benefits and lessons that are obtained from this study.

The closing chapter, Chapter 8 summarizes the dissertation by the concluding remarks of the previous chapters, and proposes the perspectives of the future work.

Chapter 2

Background

2.1 RNA and Structures

2.1.1 RNA

A *ribonucleic acid (RNA)* is a chain of ribonucleotides linked together by covalent chemical bonds, with each nucleotide containing a *ribose* sugar, a *phosphate* molecule, and one of the four nitrogenous *bases*: adenine(A), cytosine(C), guanine(G) or uracil(U) attached to the ribose. RNA is an ubiquitous family of large biological molecules that perform multiple vital roles in the coding, decoding, regulation, and expression of genes.

In the classical view of the so-called central dogma of biology, the messenger RNAs (mRNA), also referred to as the coding RNAs, serve as the template of the synthesis of a particular protein with the coded genetic information transcribed from the *deoxyribonucleic acid*(DNA) [Crick et al., 1970].

The RNAs that do not encode a protein are termed as the *non-coding RNA (ncRNA)*. More and more researches convey that the non-coding RNA molecules are critical components of transport, transcriptional and post-transcriptional regulation, chromosome replication, RNA processing and modification, mechanism of some diseases and other fundamental biological functions [Mattick and Makunin, 2006]. In the domain of RNA genomics of ribonomics, the efforts are devoted to find the determination of the physiological roles of RNA structures [Bourdeau et al., 1999].

2.1.2 RNA Structures

In the RNA world, much of the primary sequence is unimportant to function as long as the conformation and overall stability of the structure is maintained [Brierley et al., 2008]. And it is illustrated that the structural space is vastly smaller than the nucleotide sequence space [Gan et al., 2003; Haslinger and Stadler, 1999]. This means that there are a number 4^n of RNA sequences of length n theoretically, and the number of secondary structures without isolated base pairs is significantly smaller than 2^n [Grüner et al., 1996; Haslinger and Stadler, 1999]. This point of view suggests a way to survey the function of the RNA molecules, through the window of structures.

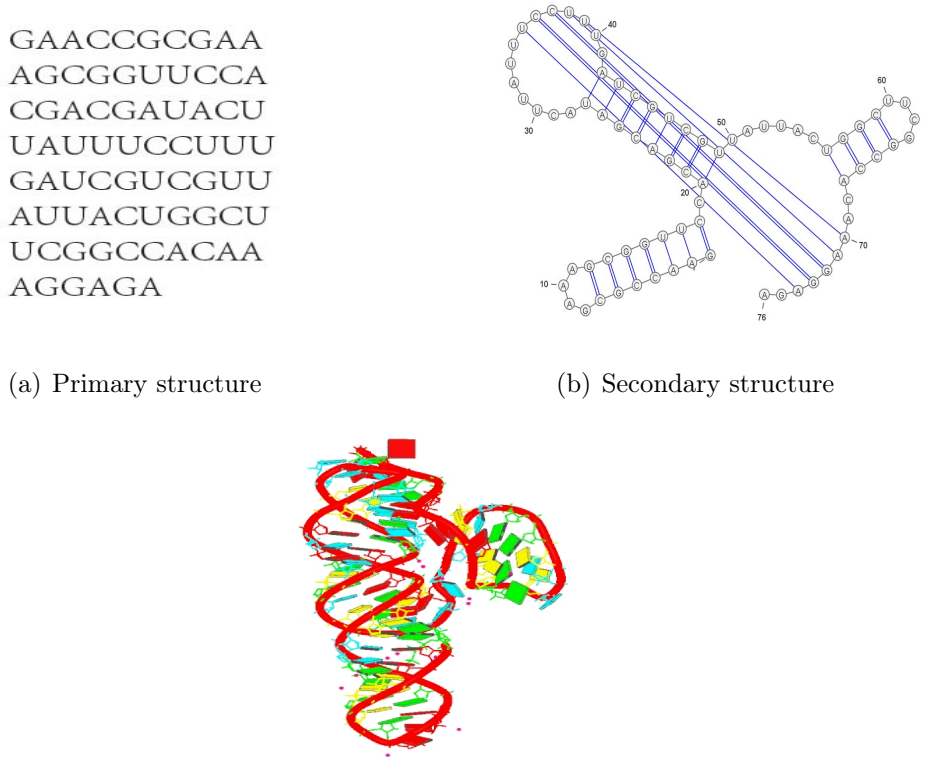
The first level of the organization of RNA structures, the *primary structure*, is the sequence of bases of the RNA chain that are attached to the sugar-phosphate backbone, and it is determined experimentally [Schmitt and Waterman, 1994; Westhof and Auffinger, 2000].

An RNA chain bends and twines about itself [Zuker and Sankoff, 1984]. Bases form chemical bonds, hydrogen bonds, with their proximal complementary neighbors, which are characterized as base pairs: two standard or canonical Watson-Crick base pairs of A with U and G with C, as well as a wobble base pair of G with U. This collection of base pairs is referred to as classical or regular *secondary structure* [Leontis and Westhof, 2001].

Besides the wobble pair, a wide variety of the non-Watson-Crick pairs, involving 30-40% of bases, contribute to the *tertiary structure* of an RNA superiorly, which organizes the loops and distortions folded up from the secondary structure precisely in space. The three-dimensional arrangement of the atoms in the tertiary structure can be decomposed into a collection of spatial interactions, some of them being promoted by spatial motifs that are held together by pairwise interactions. The tertiary structure is the level of conformation relevant for the biochemical function of the structured RNA molecule. [Tinoco Jr and Bustamante, 1999; Westhof and Auffinger, 2000; Zuker and Sankoff, 1984]

Tertiary RNA-RNA and quaternary RNA-protein interactions are mediated by RNA motifs, defined as recurrent and ordered arrays of non-Watson-Crick base-pairs [Leontis and Westhof, 2002].

An example of three levels of RNA structures is shown in Figure 2.1, where the secondary structure is visualized by *VARNA* [Darty et al., 2009], the tertiary structure is downloaded from the PDB website.



(a) Primary structure

(b) Secondary structure

(c) Tertiary structure

Figure 2.1: The hierarchical structures of *Class II PreQ1 Riboswitch RNA of Lactobacillus Rhamnosus (PDBID: 4JF2, chain A)*.

During the last decades, lots of efforts have been spent on investigating the full spatial functional conformations, the tertiary structures, with a variety of experimental techniques, such as X-ray crystallography and nuclear magnetic resonance(NMR). However, since they are extremely costly and time consuming, an alternative avenue which takes advantages of the bioinformatic methods is desired complementarily.

On the other hand, the RNA structure is declared to be hierarchical and its folding is sequential [Tinoco Jr and Bustamante, 1999] during the investigation of the tertiary structure. This survey [Tinoco Jr and Bustamante, 1999] implies that the information in the sequence flows sequentially, first to the secondary folding and then to the tertiary structure, since some secondary structures are

found to be present in the tertiary structure. The folding of RNA molecule is concluded as that the primary sequence determines the secondary structure which, in turn, determines its tertiary folding, whose formation alters only minimally the secondary structure. Meanwhile, the secondary structure is more stable than the tertiary folding and can exist and be stable independently of its tertiary folding, since the energies involved in the formation of secondary structure are larger than those involved in the tertiary interactions.

Thanks to these hypotheses, one of the most contemporarily prevalent investigations of the RNA functions increasingly focuses on the secondary structure characterization in the bioinformatic fashion, such as the theoretical prediction with the computational assistance, which provides an attractive alternative to the empirical discovery of RNA secondary structure. Computational prediction of the RNA secondary structures is the main interest of this dissertation and will be elaborated in detail in the next chapters.

2.2 RNA Secondary Structures

The secondary level of RNA structure identifies both the canonically base-paired regions as helix stems, and non-paired regions as loops. [Hendrix et al., 2005]

2.2.1 Preliminaries

From a computer science point of view, an RNA sequence S, composed of N nucleotides, can be represented as a string over the base alphabet $\{A,C,G,U\}$: $S = S_1S_2S_3...S_n$, where the sequence is numbered from 1 to n from the 5' terminus to the 3' terminus, with the S_i denoting the base corresponding to the ith position in S.

Normally, we may put more attention to the integers of positions, rather than the composition of the RNA sequence in the context of secondary structure. An RNA sequence S can be notated as a string of $[1, n]$. And in such case of prediction issues, we often focus on a partial sequence, which is termed as *fragment*. A fragment $[i, j]$ refers to the substring of S from i to j.

Two complementary bases i and j may form a base pair (i, j). Each such pair of integers represents the pairing of the ith nucleotide in S with the jth one. Normally, an *hairpin constraint* [Jiang et al., 2010] that $j - i > 3$ is taken into account, indicating that there are at least three other nucleotides in the sequence between i and j.

Given two base pairs (i, j) and (k, l), with $i < j$ and $k < l$, they can be:

- *Nested* if either $i < k < l < j$ or $k < i < j < l$.
- *Sequential* if either $i < j < k < l$ or $k < l < i < j$. Two sequential base pairs are referred to as two independent structural elements in this dissertation unless otherwise noted.
- *Crossing* or *overlapping* if either $i < k < j < l$ or $k < i < l < j$.

The *consecution* of two base pairs is a special case of the nesting, if either $k = i + 1$ and $l = j - 1$ or $i = k + 1$ and $j = l - 1$. Two consecutive base pairs form a base pair stacking.

An RNA secondary structure is an union of disjoint base pairs, where each base participates in at most one base pair. A secondary structure is *standard* or *pseudoknot-free* if all the base pairs in the structure are either nested or sequential, which are referred to as the *consistency* of the base pairs in [Stadler and Haslinger, 1997]. A secondary structure contains a *pseudoknot* if at least two base pairs are crossing.

2.2.2 Standard Secondary Structures

The standard secondary structure corresponds to a network of structural elements such as *hairpin loops*, *interior loops*, *bulges*, *multi-loops* which are also referred to as *junctions* [Gan et al., 2003; Hendrix et al., 2005] and *helical stems*. A concise definition of the elements is as follows [Andronescu et al., 2003; Spirollari et al., 2009]:

- Hairpin loop: a loop which contains exactly one base pair.
- Interior loop: a loop which contains exactly two base pairs, referred to as *internal loop* in some literatures.

- Bulge: a bulge is a special interior loop. A bulge has two base pairs but with only one side of the loop having one or several unpaired bases.
- Multi-loop: a loop which contains more than two base pairs.
- Helical stem or *helix*: a set of consecutive base pairs.
- External base: a unpaired base not contained in any loop.

Figure 2.2 shows the examples for each type of the structural elements, where the full circles in the line represent the backbone of the sequence, and the dashed lines represent the base pairs.

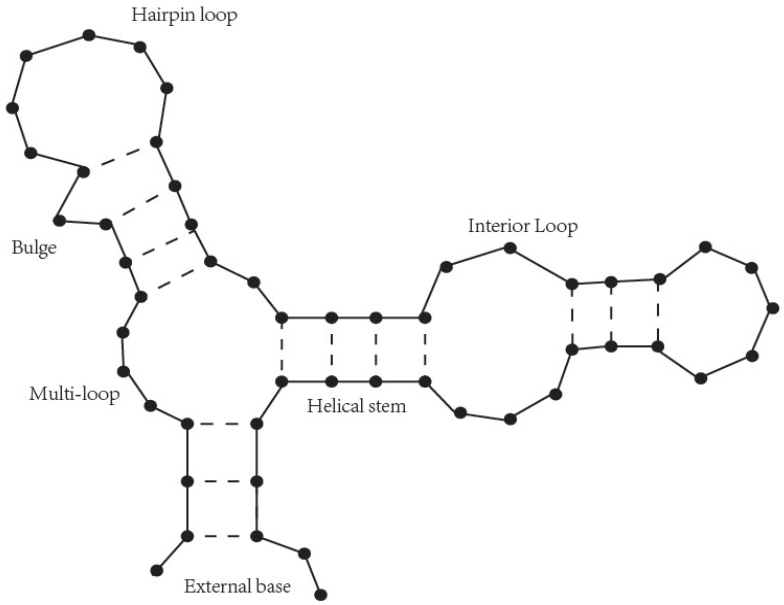


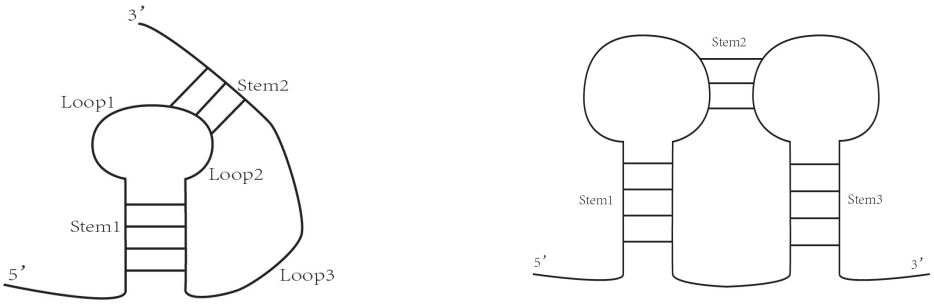
Figure 2.2: The structural elements of an RNA secondary structure.

2.2.3 Structures With Pseudoknots

The first RNA secondary structure known as a *pseudoknot*, a *hairpin-type (H-type)* pseudoknot, was found in the *turnip yellow mosaic virus (TYMV)* [Rietveld et al., 1982]. The H-type pseudoknot is formed when the single-stranded region of a hairpin loop base pairs with complementary bases outside that loop. The formation of H-type pseudoknots is known as the simplest way of forming a

pseudoknot [Ten Dam et al., 1992], and consequently ensuring them as the best characterized pseudoknots.

The base pairs in a pseudoknot break the consecutive nesting rule in the standard secondary structure. The schematic diagram of an H-type pseudoknot is shown in Figure 2.3(a), where Stem1 and Stem2 cross each other.



(a) H-type pseudoknot (b) Kissing hairpin

Figure 2.3: The schematic diagrams of an H-type pseudoknot and a kissing hairpin.

More generally, the unpaired single-stranded loops in Figure 2.3(a), *Loop 1*, *Loop 2* and *Loop 3* can harbor local secondary structure themselves, forming a recursive H-type pseudoknot or more complex one [Akutsu, 2000; Ten Dam et al., 1992]. Particularly, the beginning and ending loops of the pseudoknot can harbor a substructure locally as well, which is referred to as a *recursive* pseudoknot in this dissertation and will be elaborated more in Chapter 5. In contrast, the 3D conformation may change in the cases where one of the three loops reduces to the length of zero nucleotide. For example, in the case of the absence of *Loop 2*, the two stems become adjacent and may be stacked coaxially to form a quasicontinuous double helix.

Another prevalent type of pseudoknot is the *kissing hairpin*, formed when the unpaired bases in a hairpin loop base pair with complementary unpaired bases in another hairpin loop [Brunel et al., 2002], as shown in Figure 2.3(b). Similarly, the single-stranded loops in the kissing hairpin can harbor local substructures as well, to form a recursive pseudoknot.

The detailed introduction of the pseudoknot types and respective examples are shown in Section 5.3.1, with respect to the crossing interactions of the base pairs.

Lots of researches reveal that pseudoknots play vital roles in a variety of molecular processes, especially in viral genomes, due to the variation of structural diversity caused by the length of the loops and stems, as well as the type of interactions between them [Brierley et al., 2007; Staple and Butcher, 2005].

The functional versatility of pseudoknots includes: being involved in the recoding events such as *programmed ribosomal frameshifting* due to their more stable conformation than an equivalent hairpin, which will be introduced in detail in Chapter 4; offering binding sites for proteins or single-stranded loops of RNA; facilitating long-range interactions [Brierley et al., 2007]; maintaining the activity of telomerase [Staple and Butcher, 2005]; stabilizing the compact tertiary structures; switching the conformational states of the RNA [Ten Dam et al., 1992] and etc.

2.3 RNA Representations

From the perspective of computer scientists, some formal representations of the RNA secondary structures are desired. We can describe an RNA secondary structure, with or without pseudoknots, by some particular file formats and graphical representations. In the context of RNA pseudoknots, the *pseudoknot pattern* is quite useful to represent the crossing interactions inside the pseudoknotted conformation.

2.3.1 RNA File Formats

RNA file formats are designed so as to be able to hold the sequence data and other information about the sequence, such as the hierarchical structures.

As a preliminary, FASTA(.fasta) format is one of the most prevalent formats of sequence, and sometimes also referred to as the Pearson format, who is the author of the FASTA program [Pearson and Lipman, 1988]. The first line of a FASTA file, is a title consisting of the ID name of the sequence, starting with a '>'. Subsequent lines are composed of the sequence as a continuous string of characters from the 5' terminus to the 3' end.

The FASTA format is a well accepted type of the input for certain programs of predicting RNA secondary structures. However, on the other hand, the programs

develop particular file formats for their output, representing both the sequence and the predicted structure.

Dot-Bracket

Dot-bracket format is also referred to as *Dot Bracket Notation*, which is employed by Vienna web server [Hofacker, 2003]. The dot-bracket notation is the dominant format of the secondary structures that is adopted in the following sections of this dissertation.

In this format, the sequence is provided in the first line from 5' to 3' end, and a secondary structure with corresponding positions is given in the second line, where an unpaired base in the structure is denoted with a dot, and a base pair is denoted with a pair of opening and closing brackets.

In the standard RNA secondary structure, the base pairs are well nested, in which the opening bracket denotes the upstream 5' partner, and the closing bracket denotes the downstream 3' partner. The dot-bracket notation of 3IZF with chain C is shown in Figure 2.4(a).

However, when pseudoknots are allowed, more types of brackets are used to represent the non-nested knotted secondary structures. In the extended dot-bracket format, squared brackets, curly brackets and even alphabetical letters are employed to represent higher levels and more complicated interactions. The dot-bracket notation of 4JF2 with chain A, a typical H-type pseudoknot, is shown in Figure 2.4(b), where the squared brackets are utilized to represent the overlapping stems.

Please refer to the coming introduction of the planar and linear representations of RNA secondary structures for a better understanding of the dot-bracket notation. Quite remarkably, the Figure 2.6(a) shows the corresponding planar representation of 3IZF with chain C, the one without pseudoknot, Figure 2.6(b) shows the corresponding planar representation of 4JF2 with chain A, the one with pseudoknot. And Figure 2.7 shows the linear representations of the two genes.

BPSEQ

BPSEQ(.bpseq) format has originated from the Comparative RNA Web site [Cannone et al., 2002], storing the information of secondary structure in three columns.

```
3IZF: Chain C
GGUUGCGGCCAUAUCUACCAGAAAGCACCGUUUCCCGUCCGAUCAACUGUGUUAAGCUGGUAGA
GCCUGACCGAGUAGUGUAUGGGUGACCAUACGCGAAACUCAGGUGCUGCAAUCU
(((((((((....(((((((......((.(((...(....)...)))...)).....)))))))
.(((((.......(((((((((....)))))))))).....)))))..)))))))).
```

(a) 3IZF with chain C

```
4JF2: Chain A
GAACCGCGAAAGCGGUUCCACGACGAUACUUAUUUCCUUUGAUCGUCGUUAUUA
CUGGCUUCGGCCACAAAGGAGA
(((((((....)))))))..(((((((((.......[[[[[..)))))))))).....
.((((....))))..]]]]]..
```

(b) 4JF2 with chain A

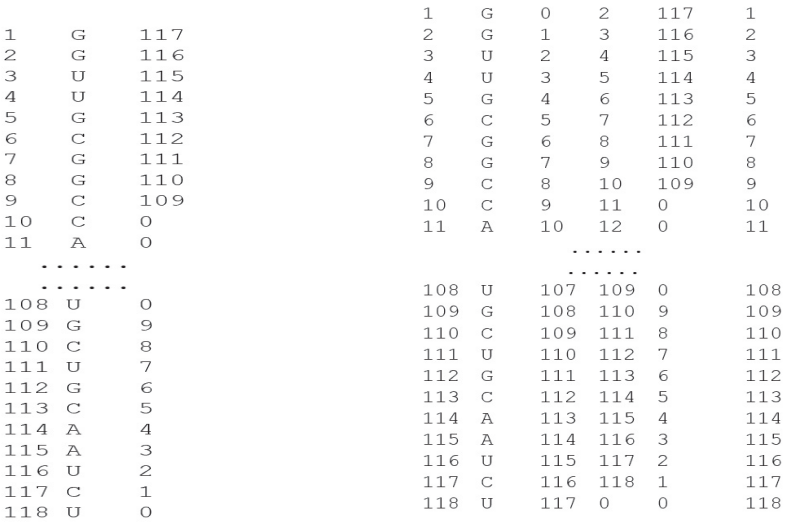
Figure 2.4: The dot-bracket notation of a standard secondary structure and an H-type pseudoknot.

The first column is the numeric positions of the sequence from 5' to 3' end, counting from 1. The second column stores the information of bases, letter by letter. The third column is the numeric position of the pairing partner of the base if it is paired, or 0 if unpaired. A part of BPSEQ file of 3IZF with chain C is shown in Figure 2.5(a).

CT

CT(.ct) format is also referred to as connect format, which has been introduced by Zuker's mfold program [Zuker, 2003]. It always stores the information of the secondary structure in six columns. The first column is the numeric positions of the sequence from 5' to 3' end, counting from 1. The third, fourth and sixth columns repeat the numeric positions again, counting from 0, 2 and 1. The second column is the sequence of bases, letter by letter. And the fifth column is the numeric positions of the pairing partner of the base if it is paired, or 0 if unpaired.

There are exceptions of the numeric value of the third and fourth columns.



```
1    G    117
2    G    116
3    U    115
4    U    114
5    G    113
6    C    112
7    G    111
8    G    110
9    C    109
10   C    0
11   A    0
   ......
   ......
108  U    0
109  G    9
110  C    8
111  U    7
112  G    6
113  C    5
114  A    4
115  A    3
116  U    2
117  C    1
118  U    0
```

(a) BPSEQ file

```
1    G    0    2    117    1
2    G    1    3    116    2
3    U    2    4    115    3
4    U    3    5    114    4
5    G    4    6    113    5
6    C    5    7    112    6
7    G    6    8    111    7
8    G    7    9    110    8
9    C    8    10   109    9
10   C    9    11   0      10
11   A    10   12   0      11
          ......
          ......
108  U    107  109  0      108
109  G    108  110  9      109
110  C    109  111  8      110
111  U    110  112  7      111
112  G    111  113  6      112
113  C    112  114  5      113
114  A    113  115  4      114
115  A    114  116  3      115
116  U    115  117  2      116
117  C    116  118  1      117
118  U    117  0    0      118
```

(b) CT file

Figure 2.5: A part of BPSEQ file and CT file of 3IZF with chain C.

For each sequence, the third column is 0 for the first base of the sequence and the fourth column is 0 for the last base. This is particularly useful to distinguish the boundaries of several sequences explicitly when they are provided in one CT file. A part of CT file of 3IZF with chain C is shown in Figure 2.5(b).

Others

There still are some formats of RNA files that store the information of RNA tertiary structure, in addition to the sequence and secondary structure.

PDB format [wwPDB, 2014]: This is an standard representation provided by the *Protein Data Bank* [Berman et al., 2000] for macromolecular structure data derived from X-ray diffraction and NMR studies. A PDB file stores various data concerning the three-dimensional structure of a molecule, the experiment carried for structure determination, the authors, etc..

RNAML format [Waugh et al., 2002]: This is an XML format that has been designed specifically to easily express data on RNA sequence and structure, allowing for the storage and the exchange of information about RNA sequence, secondary and tertiary structures. RNAML permits the description of more information about the base pairs, base triples, and pseudoknots etc.

Additionally, several RNA file formats are in existence for the purpose of mul-

tiple sequence alignment and other bioinformatical studies [course, 2014], such as Aln format, Stockholm format. They are beyond the scope of this dissertation.

2.3.2 Graphical Representations

Despite the diverse files that store the information of RNA structures, graphical representations are desired in existence for an eyeball and more intuitive comparison. Drawing an RNA secondary structure in a two-dimensional way is more aesthetically pleasing, easier to grasp and evaluate, making it the prevalent visualization of RNA secondary structures [De Rijk and De Wachter, 1997].

This part is going to introduce some classical graphical representations. They are the *planar* graph representation, the *linear* representation, and the *circular* representation, where the former two representations are massively preferred in this dissertation. Specifically, VARNA [Darty et al., 2009] utilizes all these three representations to visualize an RNA secondary structure, suggesting it as the main visualization tool of the RNA secondary structures in this dissertation.

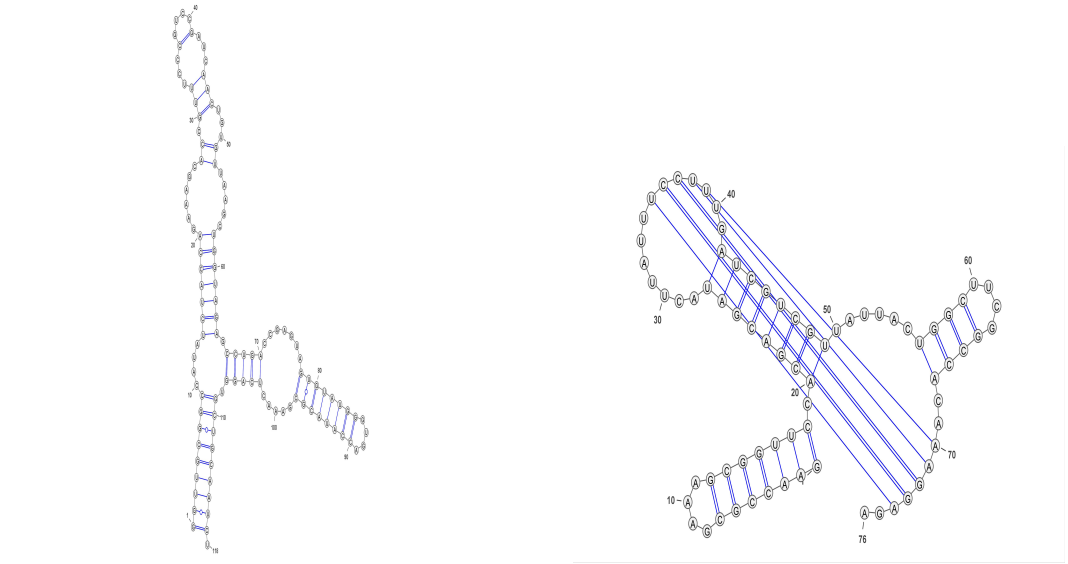
Planar Graph Representations

The planar graph representation is a conventional drawing of RNA secondary structures. The planar representation is also referred to as *rod-and-loop* representation in [Rødland, 2006]. A standard secondary structure and a pseudoknot represented in the planar graphical way are shown in Figure 2.6.

Linear Representations

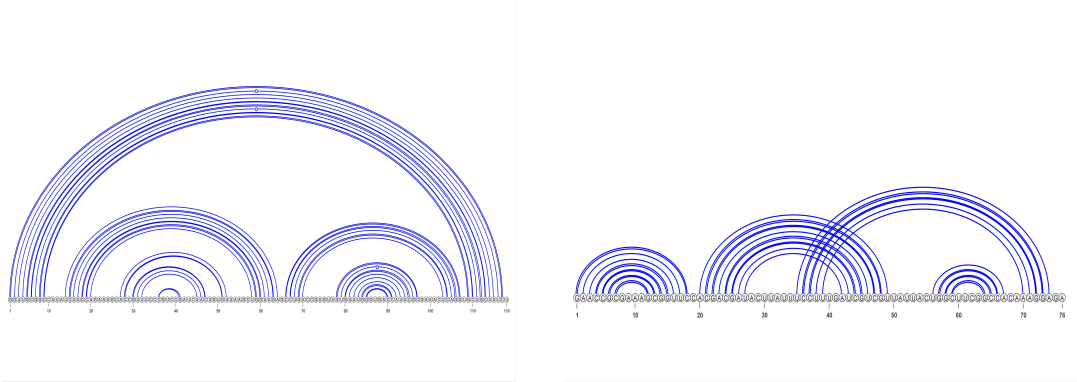
An RNA secondary structure can be represented in a linear way, which is also referred to as *bond* representation in [Rødland, 2006]. The linear representation is the dominant graphical representation in the following elaborations of RNA secondary structures.

In this representation, the RNA sequence is drawn as the backbone on a horizontal straight line, the paired bases are connected with arcs in the upper semiplane. The arcs can intersect when the pseudoknots are allowed. Examples of a standard secondary structure and a pseudoknot are shown in Figure 2.7.



(a) 3IZF with chain C (b) 4JF2 with chain A

Figure 2.6: The planar graph representations, drawn by VARNA.



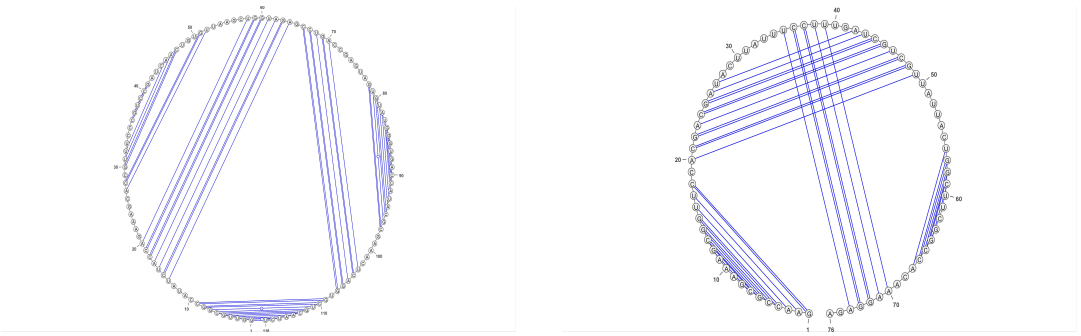
(a) 3IZF with chain C (b) 4JF2 with chain A

Figure 2.7: The linear representations, drawn by VARNA.

Others

Besides the planar and linear representations, an RNA sequence can be drawn as the backbone on a circle in the *circular* representation, with the corresponding interacting partners connected with chords. The chords can cross the others when the pseudoknots are allowed. Examples of a standard secondary structure and a pseudoknot are shown in Figure 2.8.

There are still some graph theoretical methods to visualize an RNA secondary structure, such as the *dual graphs* [Gan et al., 2003], which are available to represent both the pseudoknot-free structures and the pseudoknots. On the other



(a) 3IZF with chain C (b) 4JF2 with chain A

Figure 2.8: The circular representations, drawn by VARNA.

hand, the *RNA trees* [Fontana et al., 1993; Gan et al., 2003; Kim et al., 2013; Schmitt and Waterman, 1994; Shapiro, 1988; Zhang and Shasha, 1989] and the *forest* [Hochsmann et al., 2003] are only capable to show the pseudoknot-free structures contrastively.

2.3.3 Pseudoknot Pattern

Aiming at demonstrating the crosswise interactions or overlaps between the stems in a given pseudoknot intuitively, the *pseudoknot pattern* is used prevalently, which is defined formally by [Condon et al., 2004].

In the pseudoknot pattern representation, an even number of alphabetical letters is employed with two identical ones representing the base pairs. For example, an H-type pseudoknot has a pseudoknot pattern of $ABAB$, where two stems denoted as AA and BB cross each other. A kissing hairpin has a pattern of $ABACBC$, where the stem denoted as BB connects the stems AA and CC but promote a global non-nested conformation at the same time.

This representation provides an easy-to-understand method for us to describe the interactions inside the pseudoknot, especially for some recursive and complex pseudoknots. An example is the pseudotrefoil [Rødland, 2006], which has the pattern of $ABCABC$ revealing the mutual crosses between the stems. As a consequence, the pseudoknot pattern has an overwhelming superiority of representing the pseudoknots in this dissertation.

Chapter 3

Previous Work

This chapter principally introduces the research status of the RNA secondary structure prediction methods, given a single input sequence. It includes the methods predicting the pseudoknot-free RNA secondary structures, and the ones predicting RNA pseudoknots.

3.1 Approaches Predicting Pseudoknot-Free RNA Secondary Structures

3.1.1 Minimizing Free Energy Approach

It has been proposed that a majority of RNAs exist naturally in their thermodynamically most stable conformations, with a minimum free energy [Tinoco Jr and Bustamante, 1999]. Similarly it has been declared that the lowest free energy structure is the most represented conformation at equilibrium [Mathews, 2006]. Such kind of theories vote the most popular method during the last decades, predicting an RNA secondary structure from a given sequence, as predicting a conformation with the *minimum free energy (MFE)*. The calculated structures with higher free energies would correspond to less stable secondary structures [Tinoco Jr and Bustamante, 1999].

Introduction

Predicting an RNA secondary structure with MFE is to sum up all the energetic stabilities of each structural element, which are provided as thermodynamic parameters. The parameters are on the basis of experimentally derived free energy parameters for the base pairs, in an empirical nearest-neighbor model where the thermodynamic contributions are from both base pairing and base stacking. [Mathews, 2006; Schroeder, 2009]

The calculation of the energetically preferable structure takes advantage of computer algorithms based on dynamic programming [Mathews, 2006], which implicitly check all possible secondary structures without generating the structures explicitly, and employ the thermodynamic free energy values as their scoring scheme.

[Mathews, 2006] describes how the two steps of dynamic programming algorithms work. In the step of *fill*, the lowest conformational free energy is determined for each possible sequence fragment starting with the shortest ones, and then for the longer fragments by using a recurrence formula. In the second *traceback* step, the MFE structure is computed with the lowest free energy calculated in the fill step.

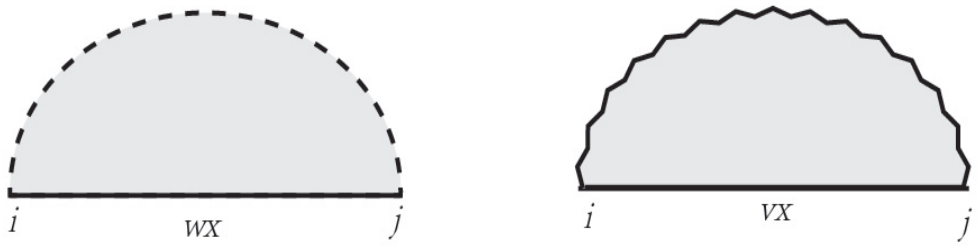
The same process is utilized by some other approaches predicting pseudoknot-free secondary structures, such as the antecedent approach proposed by Nussinov et al. [Nussinov and Jacobson, 1980] where the scoring scheme is to maximize the base pairs rather than to minimize the free energies.

Note that, the *context-free grammar (CFG)* formalism can be referred to as an alternative representation of recursions in the dynamic programming algorithms, and possibly with probabilities if the grammar is *stochastic (SCFG)*.

Zuker & Stiegler (Z&S)'s Algorithm

A pioneering approach predicting MFE structures based on dynamic programming is the *Z&S*'s algorithm [Zuker and Stiegler, 1981]. Zuker et al. propose a model of predicting pseudoknot-free structures by employing two non-gap matrices, as shown in the Figure 3.1 where the wavy line in the figures indicates that the two ends connected are definitely paired, while the dashed line indicates that

the relation between the ends connected is unknown.



(a) The '*wx*' non-gap matrix (b) The '*vx*' non-gap matrix

Figure 3.1: The non-gap matrices in the *Z&S*'s algorithm.

More precisely, given an RNA sequence of N nucleotides, the two triangular $N \times N$ non-gap matrices *wx* and *vx* in this pseudoknot-free secondary structure detecting algorithm are defined as:

- $vx(i, j)$: representing the score of the best folding between the fragment $[i, j]$ with $i < j$, provided that i and j are paired to each other; and
- $wx(i, j)$: representing the score of the best folding between the fragment $[i, j]$ with $i < j$, regardless of whether i and j are paired to each other or not.

The recursion relation used to fill the $vx(i, j)$ is given by:

$$vx(i, j) = min\ (EV1,\ EV2,\ EV3) \tag{3.1}$$

where $EV1$ indicates the energy corresponding to a *hairpin loop* that is closed by the base pair (i, j), $EV2$ indicates the energy corresponding to a *stem*, a *bulge* or an *interior loop* that is closed by the base pair (i, j), and the $EV3$ indicates the energy corresponding to a *multi-loop* between i and j, where the energy is split into the sum of two substructures, a *bifurcation* [Zuker and Stiegler, 1981]. The recursion of *vx* is shown in Figure 3.2, where contiguous nucleotides are indicated by explicit dots.

And the recursion relation used to fill the $wx(i, j)$ is given by:

$$wx(i,j) = min\ (EW1,\ EW2,\ EW3) \tag{3.2}$$

where $EW1$ indicates the energy corresponding to the case where i and j are *paired to each other*, $EW2$ indicates the energy corresponding to the case where the structure, based on $[i,j]$, has at least *one single-stranded dangling end*, namely either i or j or both do not participate in the structure, and $EW3$ indicates the energy corresponding to the case of *bifurcation* where both i and j are paired but not with each other. The recursion of wx is shown in Figure 3.3, where contiguous nucleotides are indicated by explicit dots.

The last score $wx(1,n)$ is the desired global thermodynamic score of the optimal folding, which will be used to determine a secondary structure in the traceback step.

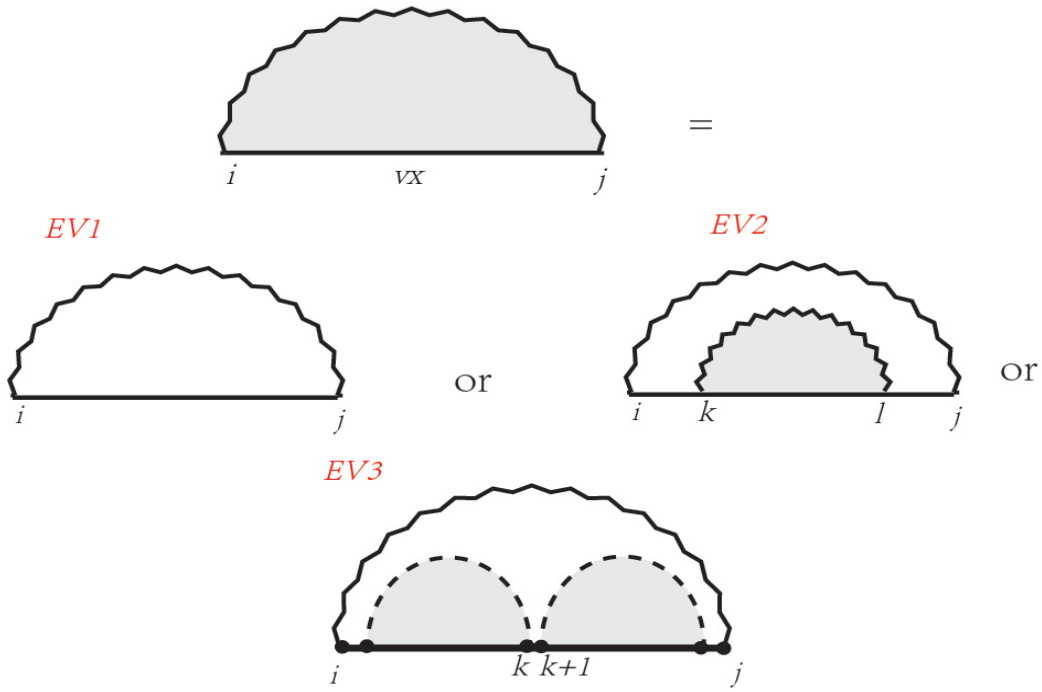


Figure 3.2: The recursion of vx in the *Z&S*'s algorithm.

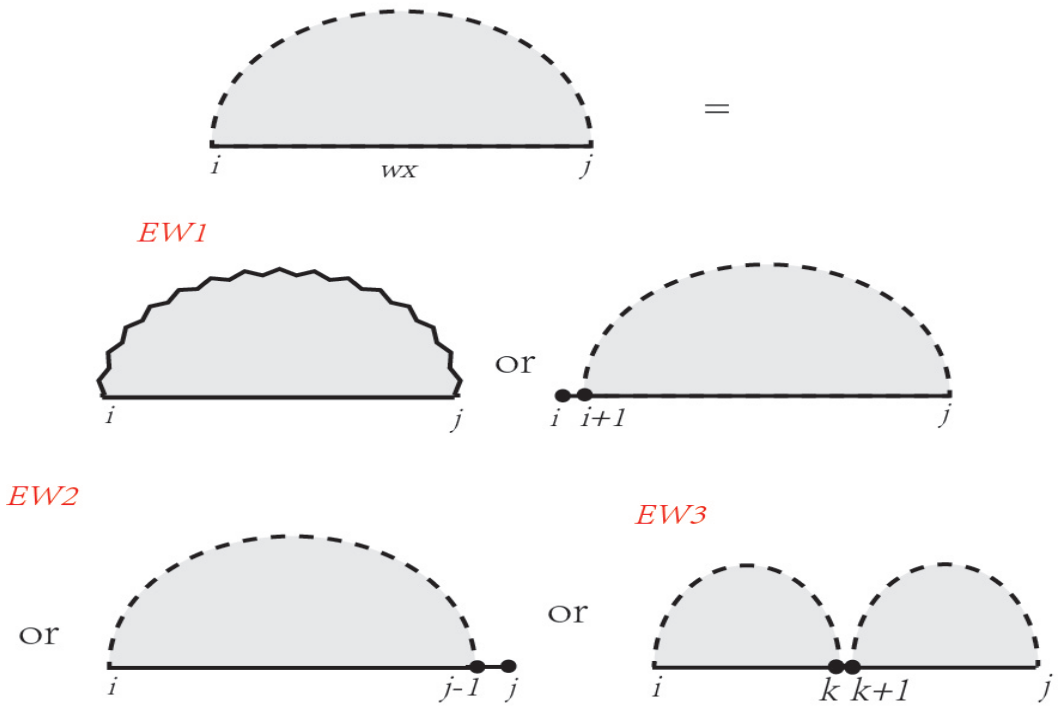


Figure 3.3: The recursion of wx in the *Z&S*'s algorithm.

Extensions

Suboptimal structures are structures that are similar in score to the structure that is predicted to have the best score. In the case of minimizing free energy, the suboptimal structures are those that have low free energies, although higher than the MFE structure. It has been pointed out that the MFE structure may not be the true structure [Ding and Lawrence, 2003], and is often not a reasonable representative for the global ensemble of secondary structures, neither single structure can be, since the structures with low free energies provide more significant information than the MFE structure [Mathews, 2006].

On the other hand, some combinations of the MFE approach and additional information, and other strategies such as covariation, phylogeny, kinetics of folding, heuristic algorithms, comparative methods, application of Bayesian statistical inference, are utilized to narrow the distribution of the ensemble of all foldings, and improve the fidelity of predicting RNA secondary structures [Ding, 2006; Zuker, 2000].

However, the accuracy of predicting RNA secondary structures by free energy minimization is limited by the incompletion of the nearest-neighbor model, the unavailability of the equilibrium state of some secondary structures, and the multi-conformations of some RNA sequences [Mathews, 2006; Seetin and Mathews, 2012]. Consequently, other considerations and improvements are proposed alternatively.

3.1.2 Statistical Approaches

Statistics refers to the thermodynamic and statistical mechanics definitions based on a Boltzmann distribution [Schroeder, 2009].

Partition Function

A partition function is a quantity that encodes the statistical properties of a system in the thermodynamic equilibrium.

The partition function Q [McCaskill, 1990] is the sum over all admissible secondary structures S of the given sequence I:

$$Q = \sum_{S \in \Omega} e^{-[F(S) \ / \ kT]} \tag{3.3}$$

where Ω is the set of all possible secondary structures for the given sequence, and $F(S)$ is the free energy of the structure S in equilibrium, and is assumed additive in terms of its loops. $F(S)$ is also referred to as ΔG^0 in other literatures such as [Mathews and Turner, 2006], and as $E(I, S)$ in the literatures such as [Ding and Lawrence, 2003]. The number k is the gas constant, and T is the absolute temperature.

Given the partition function, the Boltzmann equilibrium probability of any structure S can be calculated by:

$$P(S) = \frac{1}{Q} \, e^{-[F(S) \ / \ kT]} \tag{3.4}$$

Typically, in the set of the sampled structures, the probability of any given base pair is the frequency of its occurrence in the global ensemble of secondary structures. The probability of a given base pair (i, j) can be calculated by summing

over all the equilibrium probabilities for the structures containing that chosen binding pair, and dividing by the partition function [Mathews and Turner, 2006]:

$$P = \frac{1}{Q} \sum_{S,\ s.t.(i,j)\in S} e^{-[F(S)\ /\ kT]} \tag{3.5}$$

The calculation of partition functions and base-pairing probabilities itself, however, does not determine secondary structures [Ding and Lawrence, 2003; Mathews, 2006]. More efforts, such as *RNAstructure* [Reuter and Mathews, 2010], have been made to combine the partition function calculations with the free energy minimization to annotate the predicted minimum free energy structure with base pair probabilities from the partition function [Mathews and Turner, 2006]. Quite remarkably, the color annotation is employed to indicate the likelihood of the predicted base pairs and unpaired bases assuming the global structure in a fine-grained fashion [Schroeder, 2009].

Statistical Sampling

Based on the partition function, [Ding and Lawrence, 2003] proposes an approach to predict an RNA secondary structure by statistically sampling the structures from the Boltzmann equilibrium probability distribution of the secondary structures for a given RNA sequence. This algorithm incorporates comprehensive structural features and the thermodynamic parameters to generate a statistically representative secondary structure from the Boltzmann ensemble.

For an RNA sequence, the secondary structures in the Boltzmann ensemble are assigned with a Boltzmann equilibrium probability, which is calculated by Equation 3.4. The Boltzmann equilibrium probability distribution gives the probability for every structure, and therefore statistically characterizes the ensemble.

With the partition function $Q(1, n)$ available, the Boltzmann equilibrium probability for a secondary structure S_{1n} of sequence I_{1n} can then be computed. Under the Boltzmann model, S_{1n} is a random variable. When I_{1n} is also considered a random variable, the Boltzmann equilibrium probability is, in fact, a conditional probability of the secondary structure, given the sequence data:

$$P(S_{1n}|I_{1n}) = \frac{1}{Q(1,n)}\ e^{-[E(I_{1n},S_{1n})\ /\ kT]} \tag{3.6}$$

where the symbols are consistent to the Equations 3.3 and 3.4.

This is the scheme adopted for the secondary structure sampling algorithm described here. More specifically, given the sequence y, if we can sequentially sample x_1 from the conditional distribution $p(x_1|y)$, x_2 from $p(x_2|x_1, y)$ and x_k from $p(x_k|x_1, ..., x_{k-1}, y)$ with $k = 3, ..., m$, then $x = (x_1, x_2, ..., x_m)$ follows distribution $p(x|y)$, because the joint probability distribution is the product of the conditional distributions.

The sampling process is similar to the dynamic programming algorithms described above, but it differs in that base pairs and unpaired bases are randomly sampled with Boltzmann conditional probabilities, rather than selected by the minimum energy principle for the fragments. On the other hand, the most likely structure in a sample is the MFE structure as the probability of a structure decreases exponentially with the increasing free energy. In other words, the MFE structure has the largest sampling probability, because its Boltzmann probability is larger than that for any other structure.

Ensemble Centroid

In the sampled ensemble, [Ding and Lawrence, 2003] also proposes that the Boltzmann ensemble can be efficiently represented by distinct structural clusters, with each cluster containing similar structures.

The advantage of clustering is to find the *centroid* structure, as the single most representative of the cluster. The centroid of any set of structures is defined as the structure that has the minimum total base-pairing distance, where the base-pairing distance is the number of base pairs that differ between two structures. In other words, the centroid structure is the closest in similarity, to all the structures in the cluster. And the *ensemble centroid* is the centroid that best represents the entire collection of the structures sampled from the Boltzmann ensemble. [Ding et al., 2005]

Others

There still are some approaches to predict RNA secondary structures, such as the approach assembling a structure composed of the most probable base pairs with the maximum expected pair accuracy, which was pioneered by CONTRAfold [Do et al., 2006]. The expected accuracy is calculated by summing over both the probability of base pairs and the probability of single-stranded bases. Alternatively, in the aspect of comparative sequence analysis, as shown in Figure 1 in [Gardner and Giegerich, 2004], the approaches in this domain infer the secondary structures by determining canonical base pairs that are common among multiple homologous sequences. However, the comparative study is beyond the interest of this dissertation.

On the other hand, not all the approaches mentioned above are capable to predict pseudoknots, such as the statistical approaches based on the calculation of the partition functions [McCaskill, 1990]. The ones which are available to predict pseudoknots are elaborated in the following sections.

3.2 Approaches Predicting RNA Secondary Structures with Pseudoknots

Pseudoknots are a complex family of RNA secondary structures, whose non-nested characterization makes some of the approaches mentioned above unable to predict them. The problem of predicting RNA secondary structures including arbitrary pseudoknots, in realistic energy models, has been proved to be NP-hard [Akutsu, 2000; Lyngsø and Pedersen, 2000b; Sheikh et al., 2012]. Many approaches which are available to predict pseudoknots in polynomial time have different levels of trade-offs between the practical prediction of limited types of pseudoknots and reasonable computer cost. In other words, the approaches that are going to be introduced below consider generally a part of the pseudoknot family.

3.2.1 Exact Approaches

Exact approaches for RNA pseudoknots prediction prevalently consider the thermodynamic stability of the prediction, and/or the calculation of partition function, a maximum of base pairs etc., based on dynamic programming algorithms.

Dynamic programming is based on the observation that within optimal solutions there exist optimal solutions to smaller and self-contained subproblems. However, when pseudoknots are allowed, the broken nesting of both the structure and the energy is not sufficient to define a self-contained subproblem for the considered fragment. Thus, to use a dynamic programming algorithm for pseudoknots, simplifying assumptions about the complexity of pseudoknots must be made, as well as more intricate recursions.

Several applications to predict pseudoknots with dynamic programming are the *R&E*'s algorithm [Rivas and Eddy, 1999], extended *R&G*'s algorithm [Reeder and Giegerich, 2004], the *Akutsu*'s algorithm [Akutsu, 2000] and the *L&P*'s algorithm [Lyngsø and Pedersen, 2000a]. All algorithms search for a structure with the optimal thermodynamic stability.

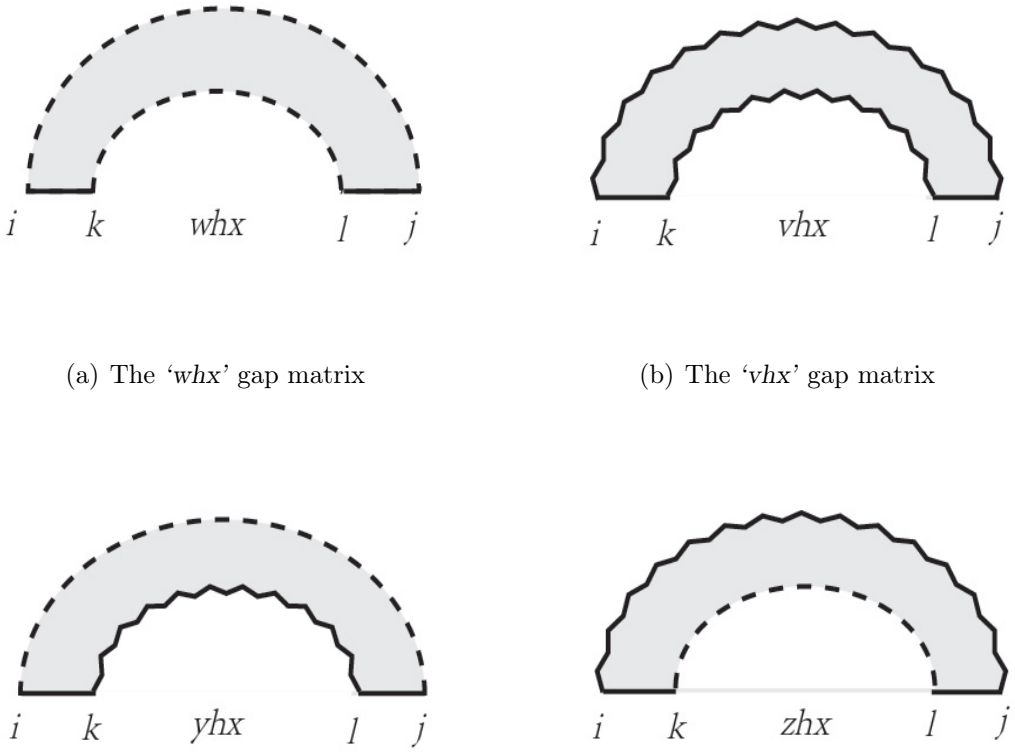
Rivas & Eddy (R&E)'s Algorithm

The *R&E*'s algorithm, sometimes referred to as the corresponding program *PKNOTS* , finds an RNA structure with minimal energy using the standard RNA secondary structure thermodynamic model, which has been pioneered by the *Z&S*'s algorithm [Zuker and Stiegler, 1981] for predicting the pseudoknot-free structures. The *R&E*'s algorithm is augmented by a few pseudoknot-specific parameters that are not yet available in the standard folding parameters, and by coaxial stacking energies for both pseudoknotted and non-pseudoknotted structures. The computer time complexity of the *R&E*'s algorithm is $O(n^6)$ and space complexity is $O(n^4)$ [Rivas and Eddy, 1999].

The implementation of the algorithm is to allow the incorporation of four gap matrices to represent the crossing conformation of pseudoknots. The non-gap matrices, utilized in detecting pseudoknot-free structures in the *Z&S*'s algorithm, are contained as a particular case of the gap matrices.

Each of the gap matrices in the *R&E*'s algorithm can in turn be constructed iteratively by the other two of those matrices, which implies that the algorithm includes in its configuration space a large variety of knotted motifs, the *R&E class* of pseudoknots.

Figure 3.4 shows the four gap matrices employed in the model of the *R&E*'s algorithm, where the wavy line in the figures indicates that the two ends connected are definitely paired, and the dashed line indicates that the relation between the ends connected is unknown.



(a) The '*whx*' gap matrix

(b) The '*vhx*' gap matrix

(c) The '*yhx*' gap matrix

(d) The '*zhx*' gap matrix

Figure 3.4: The gap matrices in the *R&E*'s algorithm.

More precisely, the gap matrices *whx*, *vhx*, *yhx* and *zhx* in the *R&E*'s pseudoknot detecting algorithm are defined as:

- *whx*: the score of the best folding that connects fragments $[i, k]$ with $[l, j]$, $i \leq k \leq l \leq j$, such that the relation between i and j and between k and l is undetermined;

- *vhx*: the score of the best folding that connects fragments $[i, k]$ with $[l, j]$, $i \leq k \leq l \leq j$, such that i and j are paired and k and l are paired as well;

- *yhx*: the score of the best folding that connects fragments $[i, k]$ with $[l, j]$, $i \leq k \leq l \leq j$, such that the relation between i and j is undetermined but k and l are paired; and

- *zhx*: the score of the best folding that connects fragments $[i, k]$ with $[l, j]$, $i \leq k \leq l \leq j$, such that i and j are paired but the relation between k and l is undetermined.

A non-gap matrix in the *Z&S*'s algorithm can be obtained by combining two gap matrices in the *R&E*'s algorithm together. In this aspect, the recursion of the pseudoknot detecting algorithm is an expansion in the number of gap matrices by adding one more case, which takes care of the crossing conformation, to the recursion of the pseudoknot-free structure detecting algorithm.

The recursion relation used to fill the $vx(i, j)$ available to predict pseudoknots is given by:

$$vx(i, j) = min\ (EV1,\ EV2,\ EV3,\ EV4) \tag{3.7}$$

where $EV1$, $EV2$ and $EV3$ are identical to Equation 3.1, while $EV4$ indicates the energy corresponding to *a non-nested multi-loop*. In detail, the *whx* connecting the fragments $[i+1, r]$ and $[k, l]$ overlaps the *whx* connecting the fragments $[k+1, j-1]$ and $[l-1, r+1]$. The recursion of *vx* in the pseudoknot detecting algorithm is shown in Figure 3.5, where contiguous nucleotides are indicated by explicit dots.

Similarly, the recursion relation used to fill the $wx(i, j)$ available to predict pseudoknots is given by:

$$wx(i, j) = min\ (EW1,\ EW2,\ EW3,\ EW4) \tag{3.8}$$

where $EW1$, $EW2$ and $EW3$ are identical to Equation 3.2, while $EW4$ indicates the energy corresponding to *a non-nested bifurcation*. In detail, the *whx* connecting the fragments $[i, r]$ and $[k, l]$ overlaps the *whx* connecting the fragments $[k+1, j]$ and $[l-1, r+1]$. The recursion of *wx* in the pseudoknot detecting

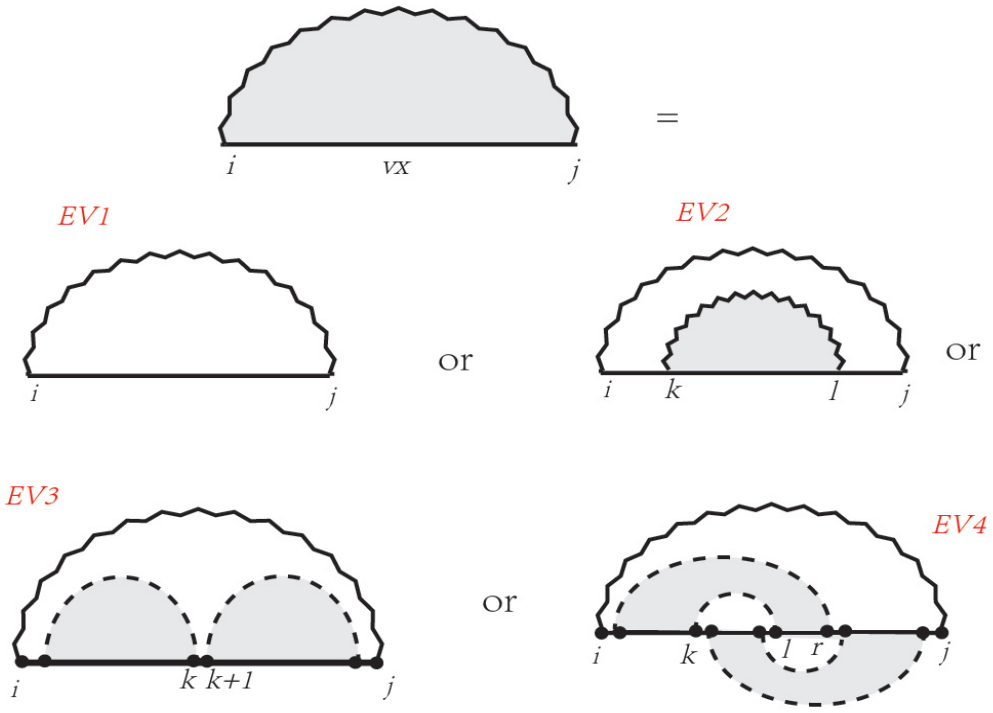


Figure 3.5: The recursion of *vx* in the *R&E*'s algorithm.

algorithm is shown in Figure 3.6, where contiguous nucleotides are indicated by explicit dots.

The algorithm can parse more complicated pseudoknots if more gap matrices are involved. But its consideration of four gap matrices is able to detect a majority of pseudoknots, both the *planar* pseudoknots and parts of *non-planar* pseudoknots, which are going to be elaborated in the classification of the pseudoknots in Section 5.3.2 and the result part of the benchmark in Section 6.1.

Analogues

There are several analogous approaches predicting pseudoknot in polynomial-time based on the dynamic programming algorithms. They adopt basically the same recurrence element with the *R&E*'s algorithm, the MFE structure on a fragment of the RNA sequence with a region restricted yet to be unpaired. However, the recurrence relations of the analogous approaches are more restricted than the *R&E*'s algorithm, making them detecting more limited types of pseudoknots, but with a lower algorithmic complexity.

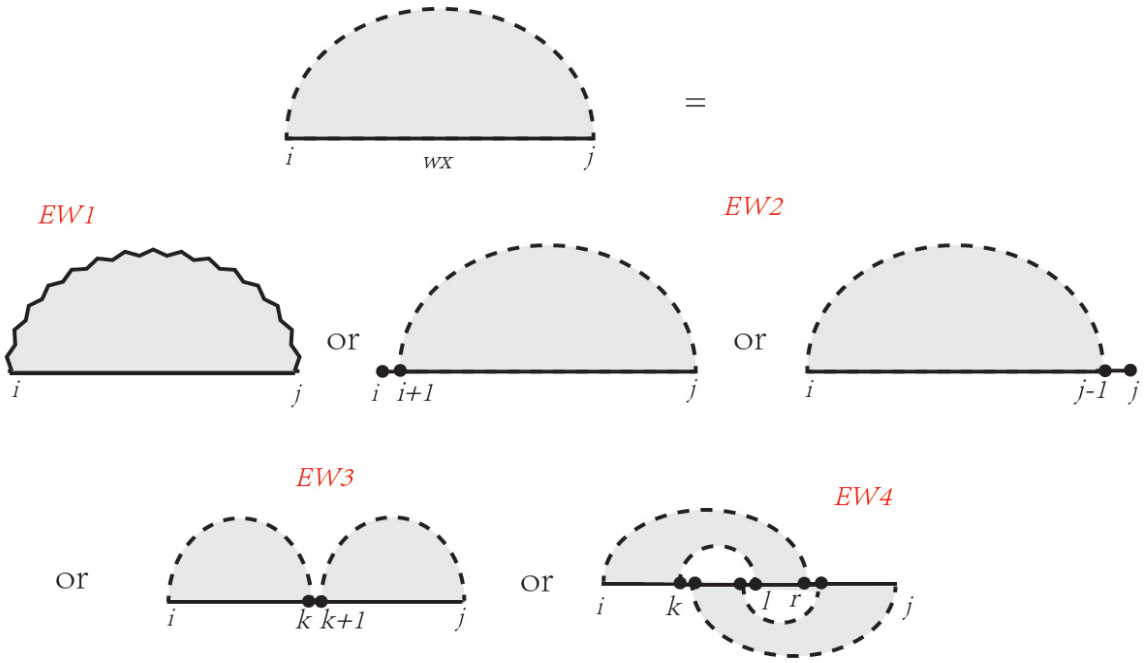


Figure 3.6: The recursion of wx in the *R&E*'s algorithm.

The *Reeder & Giegerich (R&G)*'s algorithm [Reeder and Giegerich, 2004] detects the MFE structure, by the canonization of search space of pseudoknots, and disallowing pseudoknots with more than two stems. The *Akutsu*'s algorithm [Akutsu, 2000] predicts the MFE pseudoknots which are composed of two stems. The stems are formed with the bases from three non-intersected regions of the given sequence. The *Lyngso & Pederson (L&P)*'s algorithm [Lyngsø and Pedersen, 2000a] predicts the MFE structure by summing up the energy of the computed optimal substructures based on two pairs of opposite regions of given sequence. The *Dirks & Pierce (D&P)*'s algorithm [Dirks and Pierce, 2003] calculates the partition functions of a restricted set of pseudoknots additionally.

The algorithms mentioned above take the given sequence as input, and search for the MFE structure in the search space under the respective models. In contrast, the *Jabbari & Condon (J&C)*'s algorithm [Jabbari et al., 2007], known as *HFold*, requires a pseudoknot-free secondary structure as additional input to predict the MFE structure. In detail, the *J&C*'s algorithm takes a pair of the given sequence S and a pseudoknot-free secondary structure G for S as input. The *J&C*'s algorithm

then finds another pseudoknot-free structure G' for S, which will form a density-2 secondary structure (More details about density-2 secondary structure are shown in Section 5.3.2) with G, such that the free energy of $G \cup G'$ is less than or equal to the free energy of $G \cup G''$, where G'' takes over all pseudoknot-free secondary structures for S with $G'' \neq G'$.

In conclusion, Table 3.1 shows the computational complexities of the exact algorithms mentioned above, in the order of increment of complexity, as well as the calculation models. The parameters used to predict pseudoknot-free (PKF) structures are from the algorithm *mfold* [Zuker, 2003]. More details are shown in Section 6.1 of the classification of pseudoknots.

Table 3.1: The comparison of parameters of exact approaches

Algorithm	Complexity In		Calculation Model
	Time	Space	
PKF	$O(n^3)$	$O(n^2)$	Thermodynamic Stability
J&C's	$O(n^3)$	$O(n^2)$	Thermodynamic Stability
R&G's	$O(n^4)$	$O(n^2)$	Thermodynamic Stability
A&U's	$O(n^5)$	$O(n^3)$	Thermodynamic Stability
L&P's	$O(n^5)$	$O(n^3)$	Thermodynamic Stability
D&P's	$O(n^5)$	$O(n^4)$	Thermodynamic Stability + Partition Function
R&E's	$O(n^6)$	$O(n^4)$	Thermodynamic Stability
PK	NP		

Others

The efforts that have been made to predict the pseudoknots *in silico* are not exhaustively listed in this dissertation. There are still some examples.

Besides the thermodynamic minimization, calculation of partition functions, predicting RNA pseudoknots can also maximize the number of stacking pairs under the assumption that the stacking pairs are the only loops that stabilize the secondary structures. The algorithm makes use of a geometric visualization of the planarity of stacking pairs on a rectangular grid for the approximation algorithm of the planar pseudoknots, and combines multiple greedy strategies for

the approximation algorithm of general pseudoknots [Ieong et al., 2003].

Predicting RNA pseudoknots can use such graph theoretical approach as the method detecting the structure as a number of stem sets, assembled from conserved stems across k sequences in topological order which are found by applying a *maximum clique finding algorithm* [Ji et al., 2004].

As the computational complexity and detected pseudoknots restriction of the algorithms mentioned above, the performance of the exact algorithms is often impractical, especially for long sequences and for detecting the most general types of pseudoknots. Another dilemma for some algorithms predicting pseudoknots, which are based on the energy models, is that there is little experimentally determined thermodynamic parameters for pseudoknots, making their prediction not satisfactory even for short sequences. So the coming section is going to introduce the heuristic approaches used in predicting RNA pseudoknots.

3.2.2 Heuristic Approaches

If we agree to find structures that are not necessarily with the lowest free energy, then heuristics can be applied to search for structures with low energy.

Searching a partial structure space ensures that the heuristic approaches are practical in time, and are inherently much less restricted with respect to the complexity of pseudoknot models and underlying energy models, compared to the exact approaches. But the sacrifice of the optimality of the predicted structures by the heuristic methods is unable to guarantee that they have found the global 'optimal' structure. The output of the heuristic methods is the 'best' secondary structure under their searching models.

Classical Algorithms

A *greedy* search based on Monte-Carlo simulation is proposed by [Abrahams et al., 1990]. It finds all possible stems for the given sequence, determines the free energies of their loops and base pairs, and iteratively checks the stems which will be added to the previously calculated structure with the maximum decrease of the free energy. Additionally, once a stem is added to the structure, it can not be removed in the next steps. The algorithm terminates when the MFE structure is

determined.

A *genetic* algorithm proposed by [Gultyaev et al., 1995], meanwhile, simulates the model of the RNA folding kinetics. Quite remarkably, the algorithm calculates all possible stems for a given RNA sequence, and generates the initial population of N structures. The simulation is carried out by first mutating each structure of the initial population which produces N new structures, and then crossovering the $2N$ structures and generating new population of N structures according to the *fitness*, where the fitness is defined as the total free energies of generated structures, and last increasing the chain length depending on the improvement of the free energy. The procedure of genetic algorithm simulation terminates when a predetermined number of repetitions has been done.

Others

Some other representative heuristics are as follows:

HotKnots [Ren et al., 2005], builds up the candidate pseudoknots by adding one substructure at a time to the partially formed structure, based on the thermodynamic model extended for pseudoknots as in the *D&P*'s algorithm [Dirks and Pierce, 2003].

McQFold is based on a *Markov-chain Monte-Carlo (MCMC)* method for sampling the RNA structures according to their approximate posterior distribution for a given sequence [Metzler and Nebel, 2008].

Similarly, *McGenus* [Bon et al., 2012], also uses a Monte Carlo algorithm to search for an MFE structure.

MC-Fold, is a part of the pipeline proposed in [Parisien and Major, 2008]. The main idea is based on the *nucleotide cyclic motifs (NCM)*. It infers the secondary and tertiary structures from a given sequence thanks to the empirical scoring of 3D structures.

CyloFold, is based on simulating a folding process in a coarse-grained 3D manner, and choosing stems under the established energy rules [Bindewald et al., 2010].

DotKnot, predicts the RNA pseudoknots by extracting the stem regions from the secondary structure probability dot plot, and assembling the pseudoknot can-

didates in a constructive fashion [Sperschneider and Datta, 2010].

IPknot, predicts RNA secondary structures with pseudoknots based on maximizing the expected accuracy of a predicted structure [Sato et al., 2011].

The detailed mechanisms of these methods are presented in Section 5.4.2, as they are the main considered methods of that part of work. Particularly, there are still some heuristic methods which are available to predict RNA pseudoknots. But they are excluded from the consideration, with the reasons given in Section 5.4.2 as well.

3.3 Conclusion

This chapter introduces the approaches of predicting RNA secondary structures from a single sequence, including both the standard pseudoknot-free structures and the ones containing pseudoknots. The exact methods search for the secondary structure from the entire structure space, guaranteeing an optimal output either with the minimum free energy or with the maximum statistic probability. But on the other hand, the ergodic search brings the exact methods a heavy computing burden, especially for the longer inputs, which arouses the application of the heuristics. Heuristic methods search for the secondary structure in a partial space which is reduced by the previous steps iteratively. The heuristic process costs a more agreeable complexity in time and detects a less restricted type of the pseudoknots, but with the sacrifice of the optimality of the prediction. This dissertation focuses on the prediction of pseudoknots by the state-of-the-art methods, and the comparison of their performance. These parts of work are shown in the following chapters.

Chapter 4

Frameshifting Pseudoknots and Comparison of Prediction Methods

The main interest of this dissertation is about the comparison of predicting RNA pseudoknots *in silico*, based on a variety of datasets. A series of comparisons all serve the purpose of assessing the accuracy of the predicted pseudoknots by each program. The motivation of the comparisons comes from that it is neither guaranteed nor expected that all the predictions are one hundred percent acceptable. Which of the predictions are more reliable if they conflict, especially in such situations that the reference structures which are completely determined empirically are insufficient?

In fact, comparison between predictions can not give a completely satisfactory answer. But a matching prediction between the programs can strengthen the plausibility of this prediction. On the other hand, the conflicting predictions may decrease the persuasion of either prediction, or reflect a weak spot of the prediction methods [Theis et al., 2008].

The following chapters are going to introduce a variety of comparisons principally. In this chapter, We are going to compare the prediction of *-1 programmed ribosomal frameshifting* signals by several methods. And the following three chapters are about the comparison of predicting RNA pseudoknots which are involved in much more general molecular activities.

In practice, this chapter first describes the frameshifting, one classic type of recoding events, including the motif of a frameshifting signal, the types of

frameshifting, and the frameshifting pseudoknots. Next, several programs predicting frameshifting signals are introduced.

The prediction of frameshifting signals by these methods are compared, based on three genomes. And as a significant method in this chapter, the prediction of the best predicted candidates of Orphea, which have been verified to promote frameshifting *in vivo* with a frameshifting rate over 5%, are carried out by the state-of-the-art methods additionally. Last, based on the frameshifting entries in PseudoBase, the comparison of detecting the frameshifting signals is carried out.

4.1 Frameshifting

4.1.1 Recoding events

During the expression of certain genes, the recoding events occur in response to special signals in mRNA, where two protein products are decoded from one coding mRNA [Baranov et al., 2002]. There are three main types of recoding events [Baranov et al., 2001] typically, as shown in Figure 4.1:

1), *Bypassing*, in which ribosomes suspend translation at a certain site and then resume translation downstream without decoding a block of intermediate nucleotides.

2), *Readthrough* (also referred to as *redefinition*), in which the stop codon is assigned to a different meaning, synthesizing an elongated product. Particularly the UGA can be recoded to specify the 21st amino acid selenocysteine, and the UAG can be recoded to specify the 22nd amino acid pyrrolysine.

3), *Frameshifting*, in which ribosomes switch to an alternative open reading frame (ORF) at a specific shift site. In most observed cases, the frameshifting involves a shift of one base from the reference 0 frame, either to +1 of the downstream 3' direction or to -1 of the upstream 5' direction. Some shifts of two bases are observed as well. In this dissertation, we focus on the -1 programmed ribosomal frameshifting (-1 PRF).

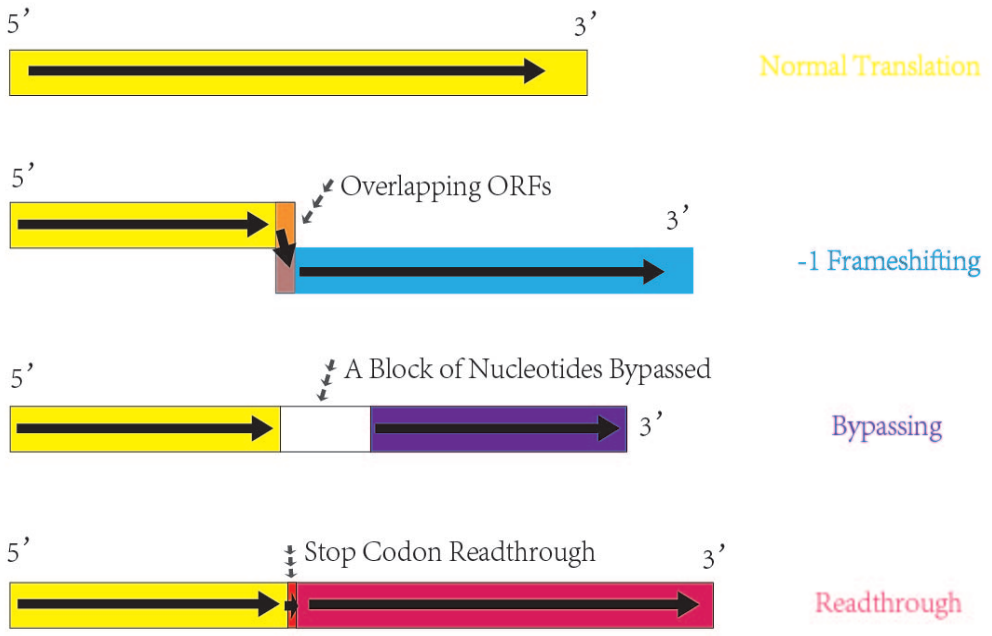


Figure 4.1: Three main types of recoding events.

4.1.2 Frameshifting Signals

-1 PRF may occur in prokaryotes and eukaryotes, and is particularly exploited by RNA viruses, with a governable ratio of efficiency [Giedroc and Cornish, 2009; Jacobs et al., 2007].

The frameshifting events are termed as programmed frameshifting since invariably important structural features of the frameshifting signals predispose the ribosome toward the shift in frames, and thus program the change. Consequently, the protein product is not directly encoded in a single ORF, but in two overlapping reading frames. Since the efficiency of frameshifting is nearly always much less than one hundred percent, this kind of recoding event allows for the expression of two primary translational products from one single mRNA that share the 5' terminal sequence encoded upstream of the shift, and differ in the 3' terminal sequence encoded downstream of the shift [Farabaugh, 1996].

-1 PRF may produce longer or shorter peptides than those synthesized from the standard decoding, thanks to the alternation of reading frames [Baranov et al.,

2002]. Figure 4.2 shows the former case where the protein synthesis starts at A, and terminates at C rather than at B, the terminal of the standard decoding.

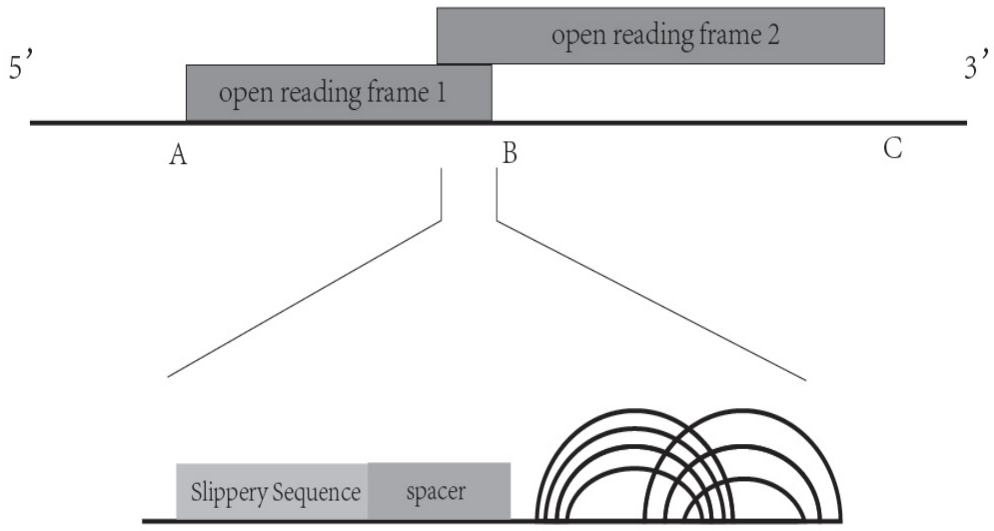


Figure 4.2: The structural elements of a frameshifting signals in the overlapping of two ORFs.

According to the common knowledge [Brierley, 1995; Brierley et al., 2007; Giedroc and Cornish, 2009], frameshifting signals contain two main elements,

- A *slippery sequence*, where the shift occurs. The slippery sequence is composed of a *heptamer* with seven nucleotides prevalently, particularly in eukaryotes and viruses, or a *tetramer* with four nucleotides [Mazauric et al., 2008], particularly in prokaryotes. The slippery sequence with seven nucleotides is referred to as $X\ XXY\ YYZ$ in the reference 0 frame, which will alter to $XXX\ YYY\ Z$ once the -1 PRF occurs. More precisely, the XXX represent three identical nucleotides, so do the YYY. And the Z can be any nucleotide. In some previous work [Brierley et al., 1992], the XXX may have some variants such as the $X_1X_2X_3$, where any nucleotide may be held by each X.

- A downstream *stimulator*, which is secondary structure adjacent to the slip-

pery sequence. An H-type RNA pseudoknot is contained more often than a stem-loop [Brierley, 1995; Brierley et al., 2007; Giedroc and Cornish, 2009].

There is some mechanical explanation on how the downstream pseudoknot stimulates the PRF [Namy et al., 2006], in which the pseudoknot blocks the mRNA entrance channel by interacting with the ribosome. But the precise mechanism of the PRF still remains as a cipher, although the pseudoknots are accepted to stimulate a frameshifting more efficiently as they are more stable to pause the ribosome [Giedroc and Cornish, 2009; Jacobs et al., 2007].

The region separating the two elements is the *spacer*, which generally contains six to eight nucleotides. There is no affirmative determination on the precise size of the spacer yet. But there exists a common agreement on the importance of this spacing distance, which must be maintained for efficient frameshifting to occur, and probably directly affects the mechanism of the frameshifting process [Brierley, 1995].

Figure 4.2 shows a frameshifting signal embedded in the overlapping region of two ORFs, where the downstream H-type pseudoknot is represented in the linear model.

The precise description of a frameshifting signal is shown in Figure 4.3, where an H-type pseudoknot follows the slippery sequence and then the spacer. Similarly to the nomenclature of [Bekaert et al., 2003], the simulating pseudoknot in the frameshifting signals is denoted as *Enhancer* in this dissertation as well. In detail, *ES1.5'* is the 5'-arm of the first stem *Stem 1*, *EL1* is the *Loop 1*, *ES2.5'* is the 5'-arm of the second stem *Stem 2*, *EL1'* is the *Loop 2*, *ES1.3'* is the 3'-arm of Stem 1, *EL2* is the *Loop 3*, and *ES2.3'* is the 3'-arm of Stem 2.

4.2 Methods Predicting -1 PRF Signals

This section is going to introduce some computer methods which have been developed for predicting -1 PRF signals, including *FSFinder*, *PRFdb*, *KnotInFrame* and *Orphea*.

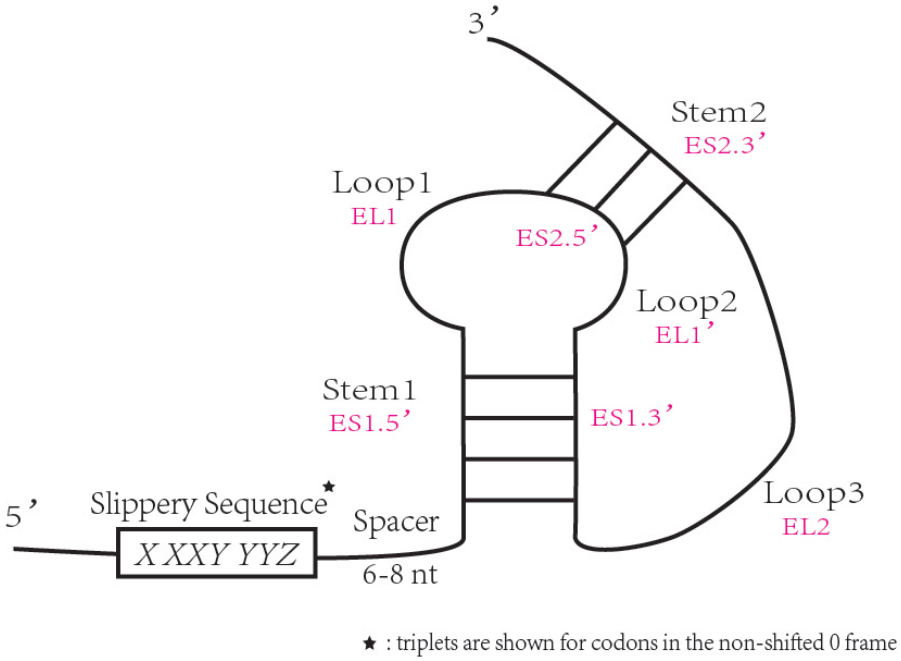


Figure 4.3: The motif of -1 programmed ribosomal frameshifting signal.

4.2.1 FSFinder

Frameshifting Signal Finder (FSFinder) [Moon et al., 2004], developed in 2004, searches the entire genome or mRNA sequences for frameshifting signals.

Specifically, FSFinder is designed to find -1 frameshifting signals, with a heptamer as the slippery sequence, in viruses, prokaryotes and eukaryotes, and two cases of +1 frameshifting sites in the eukaryotic and prokaryotic organisms. It considers both the pseudoknots and simple stem-loops as the downstream stimulatory structures.

In detail, FSFinder searches for possible slippery sequences in the form of X XXY YYZ, in which X and Z can be any nucleotide, and Y can be A or C. After a slippery sequence is identified, FSFinder searches for a downstream structure by sliding 4-11 nucleotides along the spacer. Then FSFinder filters the downstream structures which are subject to certain requirements, such as the first stem of the pseudoknot must not be larger than 13 base pairs, the second stem must not be larger than 6 base pairs, and the size of first two loops may not exceed 6

nucleotides.

The mechanism of FSFinder is to focus on an overlap of the open reading frames. It is declared that the largest ORF in the overlapping frames has the highest probability of containing frameshifting signals.

As shown in Figure 4.4, the screenshot of FSFinder shows the exploration of the overlapping region. The reading frame from A to B in frame -1 and the reading frame from C to D in frame 0 partially overlap at their termini A and D, in the region denoted as E.

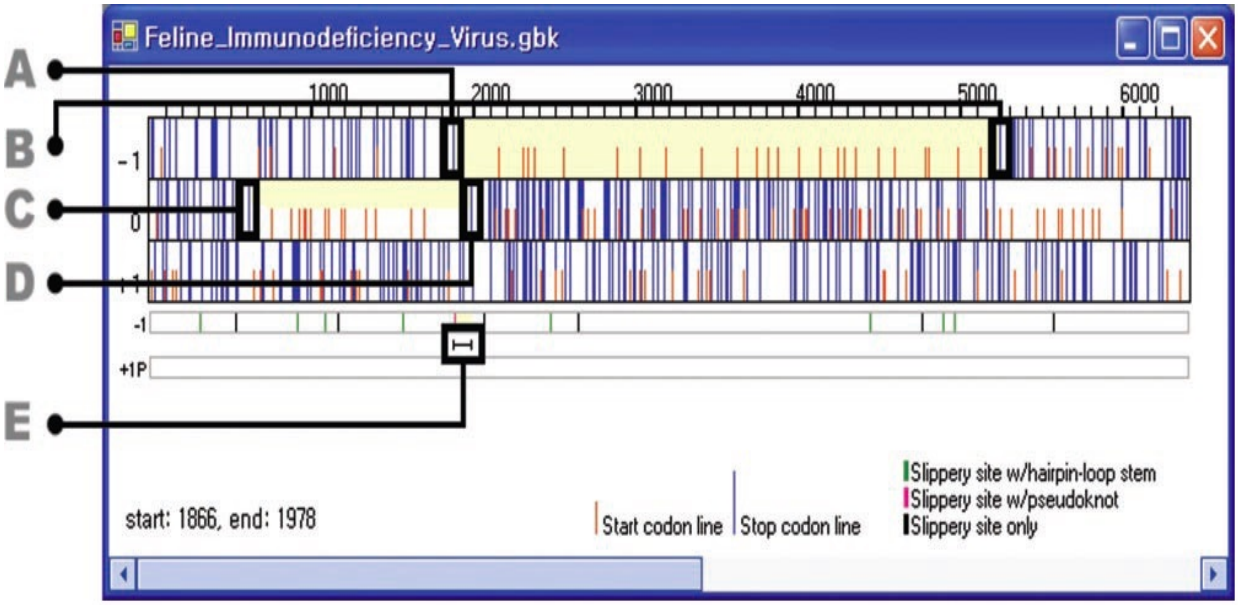


Figure 4.4: The exploration of overlapping region of FSFinder.

However, since the automatic exportation of the predictions of FSFinder is unavailable, which means that all the results are shown only in the FSFinder panels instead of a file, FSFinder must rather been used in an interactive fashion. In this aspect, the comparison of predicting -1 PRF signals by FSFinder is ignored in this dissertation.

4.2.2 PRFdb

PRFdb [Jacobs et al., 2007] is a database built in 2006, containing computer detected frameshifting candidates among which 1679 are classified as strong candidates by the authors.

Principally, the authors build an RNA motif which is learned from the analysis of viral -1 PRF signals from the RECODE database. Their aim is to find subsequences in the *Saccharomyces cerevisiae* genome and use pattern matching methods to detect the putative candidates.

In practice, *RNAMotif* [Macke et al., 2001] is employed for searching for subsequences in the genome, and the corresponding descriptor must meet several requirements. The slippery sequence is in the form of $X\,XXY\,YYZ$ in frame 0, in which X represents any identical nucleotides, Y represents A or U, and Z is not equal to G. Next, after sliding 0-12 nucleotides along the spacer, the downstream structure is searched. The allowance includes that each stem in the pseudoknot contains between 4 and 20 nucleotides in length, the first loop must be between 1 and 3 nucleotides in length, and the third loop must be at least as long as one-half the length of the first stem and not longer than 100 nucleotides.

RNAMotif then returns the collection of identified motif hits back, and pknots, the implementation of the *R&E*'s algorithm [Rivas and Eddy, 1999], is used to predict for each motif hit a MFE secondary structure.

To estimate the statistical significance and uniqueness of the predicted structure for each motif hit, the authors utilize a normalized z-score, comparing the matched candidates with those expected by chance in random genomes. This intention is because they hold the hypothesis that the frequency of finding the motif in randomized sequences can provide some insight into the likelihood that the match in natural sequence occurs by chance.

For each hit, the MFE value of the predicted structure, which is assigned by pknots, is compared to the distribution of MFE values obtained from 100 times of random shuffle and refolding of the same sequence. And the normalized z-score is given as follows:

$$Z_R = \frac{X - x}{\delta}$$

where X is the MFE value for the predicted structure, x is the estimate of the mean for the distribution of MFE values obtained from the 100 randomizations with the same sequence, and δ is the *standard deviation* of MFE values for random structures.

The -1 PRF signals having a $Z_R \leq -1.65$ and a MFE value among the lowest 25% of all the structures are considered as *strong candidate of -1 PRF signals*. These two requirements pick up those energetically strong candidates with statistically significant predicted secondary structures, which have been deposited into the PRFdb.

As a closing part, nine candidate signals possessing a wide range of feature statistics are selected for empirical testing, and further for their abilities to promote -1 PRF experimentally. More precisely, first, eight of nine candidate signals are chosen as they are predicted to fold into a pseudoknot. Next, a flexible requirement on the Z_R and MFE values of the strong signals is allowed for some of the nine candidates. The authors conclude the following:

- Every signal which contains a predicted pseudoknot promotes -1 PRF at significant levels. And the pseudoknot-free signal does not promote a measurable frameshifting.
- The frameshifting rates obtained ranges from 0.4% to 63.7%.
- Mutating the pseudoknot structure may uncontrollably affect the frameshifting. On the other hand, mutation in the spacer region can change -1 PRF efficiencies, but not completely abrogates the frameshifting.

4.2.3 KnotInFrame

With the 'dissatisfaction' of the RNAMotif that is used in the pattern matching step of PRFdb mentioned above, *KnotInFrame* [Theis et al., 2008], a similar pipeline which detects -1 PRF signals from genomic sequences, has been developed in 2008. The prime motivation of KnotInFrame comes from the declaration that most of the strong candidates with good z-score in the PRFdb do not contain pseudoknots.

Principally, the authors also build up a -1 PRF signal motif, and invoke the program *pknotsRG-fs* to predict a -1 PRF pseudoknot with the minimal free energy for the given input sequence. This program is a specialized version of *pknotsRG* [Reeder et al., 2007], which explicitly folds a given sequence into a more stable

structure than pknotsRG by modifying the grammars of the stems and loops of pseudoknots to describe a frameshifting pseudoknot more precisely.

In detail, the authors first build a -1 PRF motif with the slippery sequence in the form of $X\ XXY\ YYZ$, based on the knowledge of the frameshifting signals of RECODE, in which X represents any identical nucleotide, Y stands for either A or U, and Z for any nucleotide. The length of the spacer is between 1 nucleotide and 12 nucleotides.

Then, there are three main steps in the pipeline of KnotInFrame.

In the first *searching* phase, the pipeline scans the input sequence for occurrences of the -1 PRF motif slippery site, and folds the downstream regions by invoking pknotsRG-fs and *RNAfold* [Hofacker et al., 1994] respectively, where RNAfold returns a MFE structure without pseudoknots to the input sequence. The output of the invocation of pknotsRG-fs is notated as *pknotsRG-fs(u)*, representing the MFE value of an enforced pseudoknotted folding. The result of the invocation of RNAfold is notated as *RNAfold(u)*, representing the MFE value of an unconstrained folding without pseudoknots. And u represents the substring of the input sequence x, with the slippery sequence removed from x.

The secondary *filtering* phase has three criteria to reduce the number of candidates, based on the energy values of the constrained folding *pknotsRG-fs(u)* and the unconstrained folding *RNAfold(u)*, as follows.

The *low energy filter (LEF)* discards the candidates whose constrained energy value *pknotsRG-fs(u)* is over a threshold α, since the pseudoknots in their test are supposed to have an equal or lower energy value than the unconstrained foldings. Particularly, the authors choose the threshold of $\alpha = -7.4$ kcal/mol. The pipeline discards the candidates that are subject to:

$$pknotsRG\text{-}fs(u) > \alpha$$

Next, the *energy difference filter (EDF)* discards the candidates that rather fold into an unknotted structure, where the difference between *RNAfold(u)* and *pknotsRG-fs(u)* is larger than another threshold β. And the threshold is chosen as $\beta = 8.7$ kcal/mol. The discarded formula is:

$$RNAfold(u) + \beta < pknotsRG\text{-}fs(u)$$

The resulting set may still hold several predictions which have a same slippery site. Then, the *normalized dominance filter (NDF)* computes the length-normalized energy dominance as follows:

$$\Delta(u) = \frac{RNAfold(u) - pknotsRG\text{-}fs(u)}{|u|}$$

where Δ gives an indication of the stability of a secondary structure, namely how strong this structure outweighs the others referring to their energy values. A positive Δ means that the pseudoknotted structure is more stable than the free-folded structure. And the NDF phase only retains the candidates which maximize $\Delta(u)$.

In the third *ranking* phase, all remaining candidates passing the three filters are ranked by an evaluation function which is based on the normalized dominance of the predictions. In other words, the remaining candidates are ranked in the descending order of their $\Delta(u)$ values.

The final result of the pipeline is a list of the strongest frameshifting signals, which may have different slippery sequences, and respective structural elements and the two free energy values leading to the ranking.

4.2.4 Orphea and Ranking Process

Orphea [Brégeon et al.; Forest, 2005] is a program predicting -1 PRF signals. A ranking process follows in order to rank the predictions of Orphea.

Searching for Slippery Sequence

Orphea uses a pattern-matching algorithm to detect the slippery sequence which is in the form of *X XXY YYZ*. The algorithm then searches the candidates which have passed the requirements of the slippery motif for potential stems further. The detailed requirements of the slippery motif are considered as follows:

- $X_1X_2X_3$ must be among {*GGG, GUG, GAG, GUU, GAA, GAU, GUA, CCC, AAA, UUU*}.

- YYY must be among $\{AAA, UUU\}$.
- Z must be among $\{A, C, U\}$.
- There are two exceptions: the $G\ UGA\ AAZ$ and $G\ UAA\ AAZ$ motifs which contain stop codons in the non-shifted 0 frame.

Searching for Stimulator

For each matched slippery sequence, the program searches for a potential pseudoknot in the downstream 3' direction. A dynamic programming algorithm is employed for searching and assigning a score to each putative stem, which is the sum of scores of its base pairs.

More precisely, for the first stem ES1, ES1.5' is searched for among the first 20 nucleotides after the slippery sequence, while ES1.3' is searched for among the first 50 nucleotides.

Only the stems whose scores are above a given threshold and which satisfy some given length and distance requirements are chosen as a potential ES1. The selected ES1 stem is stepped into a similar stage for a further search of the second stem ES2. The overlap of two stems must be retained, namely the ES2.5' must lie between ES1.5' and ES1.3', and ES2.3' must lie after ES1.3'. The two stems may contain bulges.

The parameters and thresholds used by Orphea depend on the statistics which were computed on 17 known frameshifting signals in viral genomes. These 17 viral signals are referred as the *learning data* of Orphea in the following parts.

Candidates Ranking and Selecting

However, as the candidates detected by Orphea may be numerous, the authors have designed a method to rank all the candidates in order to find out the best predictions of Orphea, according to some scores which take advantage of the known frameshifting sites.

In this step, several machine-learning approaches are combined to give a rank to each candidate, and the top ranked ones are supported to be the most promising candidates of -1 PRF signal, and are considered to be tested experimentally.

In detail, the authors have trained four predictors, *J48*, *JRip*, *Random Forest* and *Naive Bayes* which are implemented in *WEKA* [Hall et al., 2009], by exploiting the features of the known frameshifting signals. The candidates having the best predicted rate within all the predictors are considered as the most promising ones. In other words, the ranking scheme here follows a consensus policy to select the candidates as the promising ones.

Compared to the third ranking phase of the pipeline in KnotInFrame, the ranking scheme for ranking the predictions of Orphea is different. Orphea ranks all the candidates which have reached the requirements of the detecting model, and the candidates may have different slippery sequences. On the other hand, KnotInFrame ranks the candidates by their $\Delta(u)$ values in the context of the candidates with a same slippery sequence. This is determined by the premise of u, the substring of the input sequence with the slippery sequence removed, as shown in Section 4.2.3.

As the most interesting work that the authors of Orphea have done, testing the propensity of the best ranked candidates to induce -1 PRF *in vivo* [Brégeon et al.] may verify the fidelity of the predictions of Orphea. This part of work differs from the research carried out by PRFdb, whose empirically testing candidates possess a wide range of feature statistics, rather than good rankings, as shown in Section 4.2.2. The second difference from PRFdb is that the authors of Orphea have tested the predictions of the -1 PRF signals by Orphea based on the human mRNAs and a synthetic genome, in addition to the *Saccharomyces cerevisiae* genome.

The work-flow of the searching step of Orphea and the ranking step is shown in Figure 4.5.

4.3 Evaluation

4.3.1 Parameters

For quantifying the accuracy of the predicted RNA secondary structures, three evaluation criteria are used in this dissertation.

- The *sensitivity*, which is called *recall* in the information retrieval community.

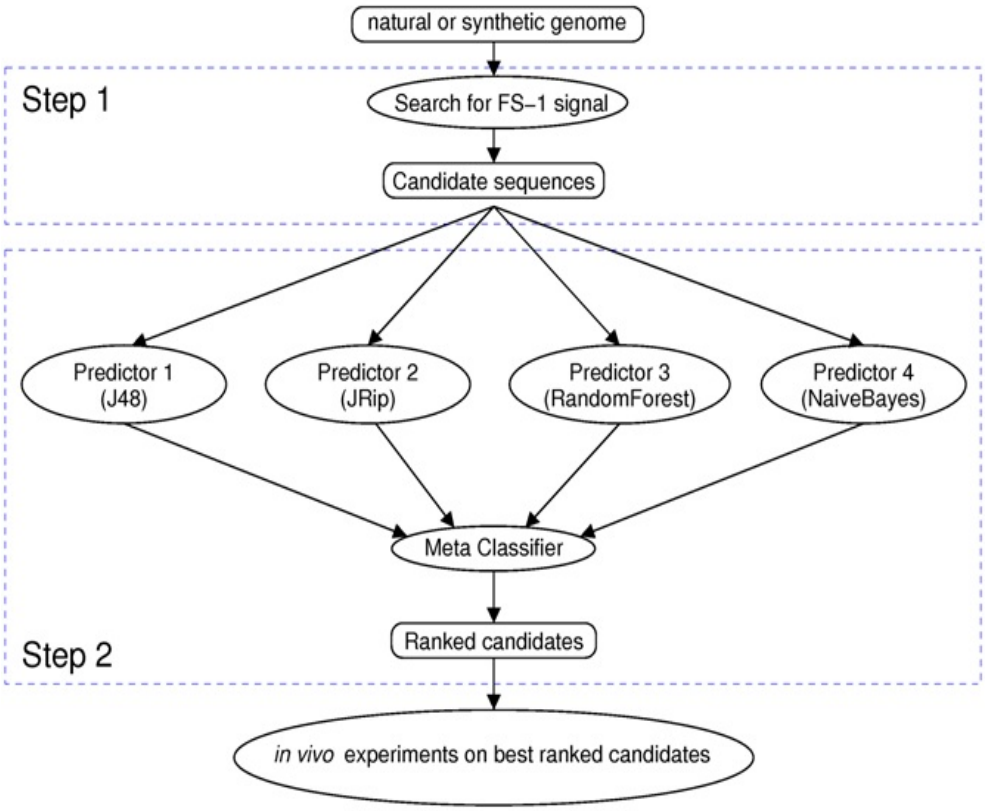


Figure 4.5: The work-flow of Orphea and ranking process, taken from Fig.2 in [Brégeon et al.].

- The *positive predictive value (PPV)*, which is called *precision* in the information retrieval community. The PPV is also referred to as the *selectivity* in [Gardner and Giegerich, 2004].
- The *Matthew's correlation coefficient (MCC)* [Matthews, 1975], which combines the sensitivity and PPV, suggesting it as a more representative and comprehensive parameter.

The formal definitions of these three measures are:

$$Sensitivity = \frac{TP}{TP + FN} = \frac{TP}{number\ of\ base\ pairs\ in\ reference} \tag{4.1}$$

$$PPV = \frac{TP}{TP + FP} = \frac{TP}{number\ of\ base\ pairs\ in\ prediction} \tag{4.2}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.3}$$

where the *positives (P)* and the *negatives (N)* both refer to base pairs. Particularly, TP is the number of *true positives*, the set of correctly predicted base pairs, FP is the number of *false positives*, the set of incorrectly predicted base pairs, FN is the number of *false negatives*, the set of base pairs in the reference structure that are absent in the predicted one, and TN is the number of *true negatives* [Puton et al., 2013], the set of correctly predicted unpaired bases.

Equations 4.1, 4.2 and 4.3 are the standard ways to calculate the sensitivity, PPV and MCC, which are prevalently employed in the assessment of predictions. And they are used to assess all the predictions in this dissertation as well.

4.3.2 Variants

One of the pioneered applications of the three parameters on the evaluation of RNA secondary structure prediction is proposed in [Gardner and Giegerich, 2004], where the positives and negatives are used to refer to the base pairs. However, their definitions are sightly different from Equations 4.2 and 4.3, with an introduction of the subtraction of ξ from the false positives.

$$PPV = \frac{TP}{TP + (FP - \xi)} \tag{4.4}$$

$$MCC = \frac{TP \times TN - (FP - \xi) \times FN}{\sqrt{(TP + FP - \xi)(TP + FN)(TN + FP - \xi)(TN + FN)}} \tag{4.5}$$

The subtraction of ξ is introduced because the authors believe that some of the FP are not equally false, assuming the FP can be classified as either *inconsistent*, *contradicting* or *compatible*.

- The inconsistent group of false positives is the set of predicted base pairs that conflict with a base pair in the reference structure, namely either end of a base pair in the reference structure has a base-pairing with another base in the predicted structure.

- The contradicting group is the set of predicted base pairs that are not nested with respect to the reference structure, namely a predicted base pair crosses

one base pair in the reference structure, and both ends of the predicted base pair are unpaired in the reference structure.

- The compatible false positives are those neutral with respect to the reference structure, namely a predicted base pair does not satisfy the two requirements above and is not present in the reference structure. Their number is denoted as ξ in Equations 4.4 and 4.5.

In practice, the compatible false positives ξ are subtracted from the false positives as the authors declare that this part of predicted base pairs does not conflict with the reference structure. The acceptance is supported by the requirements that both ends of the compatible false positives are unpaired in the reference structure, and the formed base pair does not intersect any base pair in the reference structure at the same time.

Figure 4.6 shows an example of the positive and negative predictions compared to the reference structure. Specifically, the number of base pairs in the reference structure, which is shown in the upper semi-plane, equals to the sum of the number of *TP* and *FN*. On the other hand, the number of base pairs in the predicted structure, which is shown in the lower semi-plane, equals to the sum of the number of *TP* and *FP*. *TN* is calculated by subtracting the existing pairs *TP* from all possible base pairs, where all the possible base pairs are the exhaustive number of the *A:U*, the *G:C* and the *G:U* pairs in this given sequence.

As mentioned above, this dissertation prefers to evaluate the predictions with the parameters calculated according to the Equations 4.1, 4.2 and 4.3, without further considering the subsets of the false positives. This consideration is made by noticing that, when dealing with pseudoknots, the notions of 'contradicting' and 'compatible' base pairs are irrelevant. This argument is illustrated in Figure 4.7, where the base pairs in the reference structure are shown in solid lines in the upper semi-plane, and the predicted base pairs are shown in dashed lines in the lower semi-plane. The notation of the base pairs in both structures are numbered in the order of the numeric position of their 5' ends.

Figure 4.7(a) depicts a pseudoknot-free secondary structure. We may classify the predicted base pair *P1* as a contradicting base pair because of its overlap

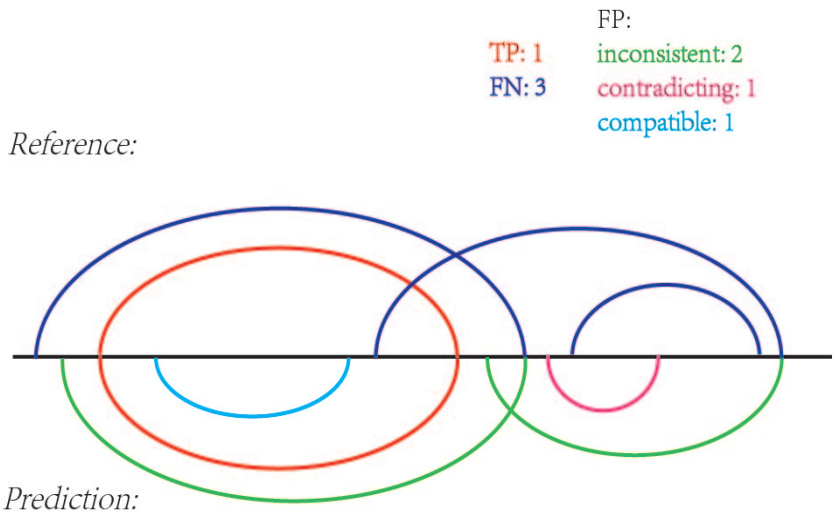


Figure 4.6: The schematic example of the positive and negative predicted base pairs.

with the base pair *R1* in the reference structure. And we may also classify the predicted base pair *P2* as a compatible one since it does not cross any base pair in the reference structure and neither end of *P2* forms a base pair in the reference structure. We notate that the base pair *P2* is *embedded* in the base pair *R2*. Figure 4.7(b) depicts a pseudoknot. The predicted base pair *P1* is contradicting with the reference structure as it crosses the *R2*. While the *P2* is compatible with the reference structure.

But it is interesting to analyze some other cases, especially the base pairs highlighted in red in Figure 4.7, where the prediction has a pseudoknotted conformation.

According to the division of false positives introduced above, the predicted base pair *P2* in the Figure 4.7(c) is a compatible false positive. And *P1* and *P3* are two true positives as they correspond to two reference base pairs, *R1* and *R2*. But we hold a different opinion on the classification of the predicted base pair *P2* in the Figure 4.7(d), which is highlighted in red. *P2* may be contracting as it crosses *R2* in the reference structure. We argue that *P2* is a 'compatible' false positive, as it is embedded in the correctly predicted base pair *P1* and does not break the global crossing interactions in the reference structure.

Let us go further. The predicted base pair *P1* in the Figure 4.7(e) is classified

as a contradicting base pair as it crosses the base pair *R2* in the reference structure. But it is the *P1* that makes the prediction a pseudoknotted conformation globally, much closer to the crossing interaction shown in the reference structure. We strongly prefer to classify *P1* as a 'compatible' base pair.

Further, we lengthen the 3' end of the compatible *P2* in the Figure 4.7(e) to make it cross the base pair *R1*, as shown in Figure 4.7(f). Is *P2* a contradicting base pair right now? We still argue that *P2* is 'compatible' with the reference structure, as it is embedded in the correctly predicted base pair *P3*.

In fact, if we examine all the base pairs in red in Figures 4.7(d), 4.7(e) and 4.7(f), we may conclude the argument of their compatibility with the reference structure is well supported by the reasons as follows:

- They cross the base pairs in the reference structure. *P2* in Figure 4.7(d) crosses *R2*. *P1* crosses *R2* in both Figures 4.7(e) and 4.7(f). And *P2* in Figure 4.7(f) crosses *R1*.

- Simultaneously, the base pairs in red are embedded in some base pair in the reference structure. *P2* in Figure 4.7(d) is embedded in *R1*. *P1* is embedded in *R1* in both Figures 4.7(e) and 4.7(f). And *P2* in Figure 4.7(f) is embedded in *R2*.

In our opinion, a more comprehensive classification of the false positives is as shown in Algorithm 1.

But unfortunately, our confidence on the classification shown in Algorithm 1 fails quickly as we encounter the examples as shown in Figures 4.7(g) and 4.7(h).

The predicted base pair *P1* in Figure 4.7(g) crosses *R2* in the reference structure and concurrently is embedded in *R1*, suggesting *P1* a compatible false positive in accordance to Algorithm 1. But on the other hand, the 5' end of *P2* forms a base pair *R2* in the reference structure, should we accept it as the inconsistent false positive rather than a compatible one? Furthermore, all the three predicted base pairs in Figure 4.7(h) are classified as compatible ones according to the Algorithm 1, should we accept them as compatible predictions?

Actually, we may classify the pseudoknot shown in Figure 4.7(g) as an acceptable prediction. But contrarily, we do not accept the prediction shown in Figure

Algorithm 1 The Classification of the False Positives.

Input: The set of false positive base pairs (x, y).

Output: The classification of each (x, y) into the *inconsistent*, *contradicting* and *compatible* subsets.

```
if (x, y) shares either end with one base pair in the reference structure. then
    (x, y) is an inconsistent base pair.
else                          ▷ Both ends of (x, y) are unpaired in the reference structure.
    if (x, y) crosses any base pair in the reference structure then
        if (x, y) is embedded in one base pair in the reference structure then
            (x, y) is a compatible base pair.
        else
            (x, y) is contradicting with the reference structure.
        end if
    else
        (x, y) is a compatible base pair.
    end if
end if
```

4.7(h) as it is a completely different pseudoknot from the reference structure.

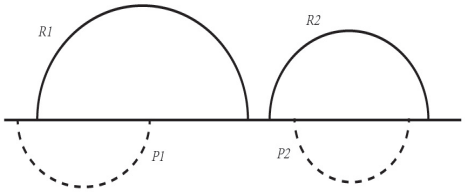
The examples shown in Figure 4.7 are not exhaustive. So, what precise definitions of classifying the false positives should we adopt?

Frankly speaking, as the unavailability of a systemic interpretation of the crossing interactions in an arbitrary pseudoknot, we can not perceive the quintessence of the similarity between the prediction and the reference structure. We may not define the degree of acceptation in the sequel. So we prefer to use the criteria calculated by the Equations 4.1, 4.2 and 4.3, without further considering the classification of the false positives, as there are no better options of evaluation to choose and no reliable suggestions on a practical applicability of them.

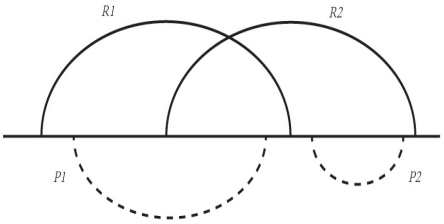
4.3.3 Why Not ROC Curve?

Someone may wonder why not evaluating the predictions by the *receiver operating characteristic (ROC)* analysis which is a graphical plot illustrating the performance.

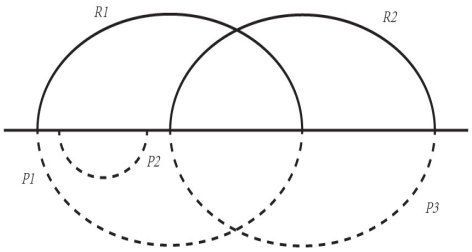
In fact, the y-axis of the ROC curve is the sensitivity, which is referred to as



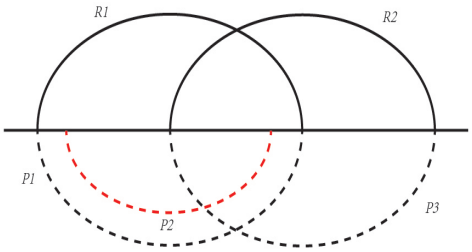
(a) Case 1



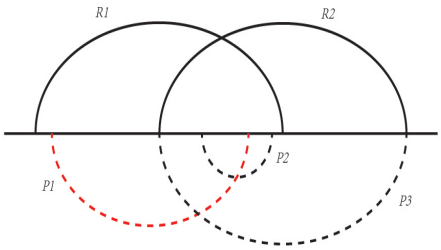
(b) Case 2



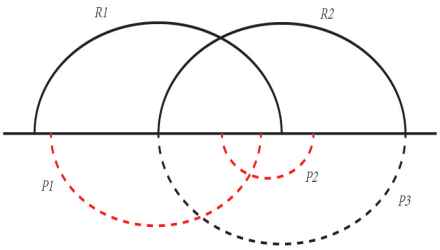
(c) Case 3



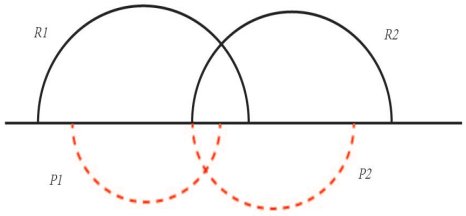
(d) Case 4



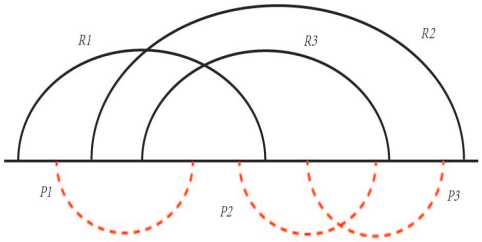
(e) Case 5



(f) Case 6



(g) Case 7



(h) Case 8

Figure 4.7: The schematic examples of the classification of false positives.

the *true positive rate (TPR)* and calculated by the Equation 4.1. And the x-axis is the *false positive rate (FPR)* which is calculated by the Equation 4.7. Each prediction given by a certain method corresponds to a point with respect to the number of their positives and negatives.

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{4.6}$$

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N} \tag{4.7}$$

However, the ROC analysis is not taken into account by this dissertation for two reasons. First, we expect each ROC curve may reflect the performance of prediction by one method. But the predictions are returned by the certain method, where the threshold of classifying of the positives and negatives may not be altered. Consequently, the ROC figure may be composed of the discrete points, rather than a classical curve passing from the point with both TPR and FPR equal to 0, to the point with both values equal to 1. Second, *TN* in Equation 4.7 is calculated by subtracting *TP* from all possible base pairs based on the given sequence. The amount of all possible base pairs induces *TN* to cover a quite large number of pairs. The enormous gap between *FP* and *TN*, which may range from ten times to thousand times, contributes FPR a quite low value approaching to 0. These low values induce a quite thicker aggregation of discrete points in the small area near the y-axis, compared to the global plane.

Obviously, the ROC analysis is not very likely to succeed in the evaluation of predicting RNA secondary structures.

4.4 Comparison of Predictions

This section introduces a series of comparisons. A comparison of the parameters that the programs mentioned above utilize is introduced first. Then, the predictions of Orphea and KnotInFrame, based on the *Saccharomyces cerevisiae* genome are compared with the strong candidates of PRFdb, and the predictions of Orphea and KnotInFrame, based on a synthetic genome and Human mRNAs are compared with each other. Additionally, predicting the best predictions of

Orphea, which have been tested empirically, is performed by the contemporary methods. Last, the comparison of the predictions of Orphea and KnotInFrame with the viral frameshifting signals in PseudoBase is carried out.

4.4.1 Comparison of Parameters

We can not deny the difference between the approaches is strongly due to the diverse definitions of the slippery sequence, the size of the downstream pseudoknot structure and some other elements of a frameshifting signal, and particularly the algorithm used to search for the structures. In this aspect, a reasonable comparison of the predictions should start from a good comprehension of the respective divergences of the parameters used by each method.

The parameters of the four programs introduced above are shown in Table 4.1, where the rightmost column *Energy* illustrates whether the thermodynamic parameters are considered by the corresponding program in their detecting models.

4.4.2 Prediction of Three Genomes

This section is about the comparison of the common predictions based on three genomes. The predictions can be referred to as common if their slippery sequence is the same at the same slippery position in the genome, regardless of the shapes of the predicted secondary structures.

Datasets

We have tried to find the common predictions of the methods, in the context of the following three datasets:

- The *Saccharomyces cerevisiae* genome, which has been obtained from Saccaromyces Genome Database as of April 2011, of length about twelve million nucleotides.
- A synthetic genome, which has been generated as a random sequence with the GenRGenS software [Ponty et al., 2006], of length of twelve million nucleotides, with the same average composition in hexanucleotides as the *Saccharomyces cerevisiae* genome (12Mb).

- Human mRNAs, which have been obtained from the 01/09/09 version of the NIH Mammalian Gene Collection, a number of 42,433 sequences with lengths ranging from dozens of nucleotides to tens of thousands of nucleotides. Typically, we refer to the Human mRNAs as the third genome without special notification for convenience.

Table 4.1: The comparison of parameters of four programs.

Program	Slippery Sites	Spacer	ES1	ES2	EL1,EL1' and EL2	Learning Data	Energy
FSFinder	X = N Y = A or U Z = N	4 - 11 nt	4 - 13 bp At least 2 G:C in first 4 bp	0 - 6 bp	EL1: 0 - 6 nt EL1': 0 - 6 nt EL2: 6 - 30 nt	RECODE[1] and PseudoBase	NO
PRFdb	X = N Y = A or U Z ≠ G	0 - 12 nt	4 - 20 /nt G:U allowed	4 - 20 /nt G:U allowed	EL1: 1 - 3 nt EL1': optional EL2: 1/2EL1 - 100 nt	56 sequences in RECODE	YES
KnotInFrame	X = N Y = A or U Z = N	1 - 12 nt	4 - 17 bp/nt	3 - 18 bp/nt	EL1: 1 - 10 nt EL1': 0 - 50 nt EL2: 6 - 40 nt	28 sequences in RECODE	YES
Orphea	$X_1X_2X_3$= {GGG, GUG, GAG, GUU, GAA, GAU, GUA, CCC, AAA, UUU} Y = A or U Z ≠ G	3 - 9 nt	The ES1.5' lies in the first 20 nt from Z. The ES1.3' lies in the following 30 nt from Z. May contain bulges.	The ES2.5' lies between ES1.5' and ES1.3'. ES2.3' is searched alternatively according to the the length of ES1.	EL2: ≥ 4 nt	17 sequences in PseudoBase	NO

[1] The number of sequences in the learning data of FSFinder in unknown.

Results

Orphea had 171 predictions based on the *Saccharomyces cerevisiae* genome, 102 predictions on the synthetic genome, and 4414 ones on the human mRNAs. All of them are available in Supplementary File *Frameshifting*.

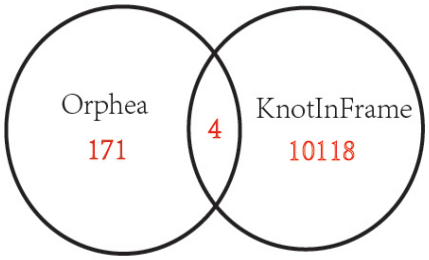
To carry out the comparisons, I practically ran KnotInFrame through its web service.

For the entire *Saccharomyces cerevisiae* and synthetic genomes, I cut the sequences into pieces of approximately 40 000 nucleotides, on the advice of the developers of KnotInFrame about the maximum length of input. This was done in an overlapping fashion to avoid the omittance of the candidates which locate potentially in the overlap of any two consecutive pieces. The pieces of sequence were sent successively as input to the web service of KnotInFrame, ensuring the candidates that locate on two different pieces can be detected. KnotInFrame also needs a parameter of the maximum number of best candidates that can be detected in the given input sequence. I fixed this number to 15, which is 10 by fault, for a larger group of 'best' predictions returned by KnotInFrame. Otherwise, I employed the default parameters. For the Human mRNAs, I sent the 42,433 sequences directly into the web service of KnotInFrame.

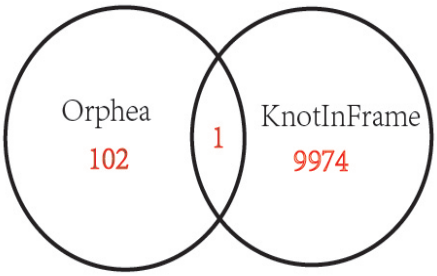
As a result, KnotInFrame had 10118 predictions based on the *Saccharomyces cerevisiae* genome, 9974 predictions based on the synthetic genome, and 160 509 ones based on the Human mRNAs respectively (available in the Supplementary File *Frameshifting*).

There were 4 common candidates found between the 171 predictions of Orphea and the 10118 predictions of KnotInFrame based on the *Saccharomyces cerevisiae* genome. Based on the synthetic genome, only 1 common candidate was found between the 102 predictions of Orphea and the 9974 predictions of KnotInFrame. And based on the human mRNAs, 70 common candidates were found between the 4414 predictions of Orphea and the 160 509 predictions of KnotInFrame. The results are shown in Table 4.2 and Figure 4.8.

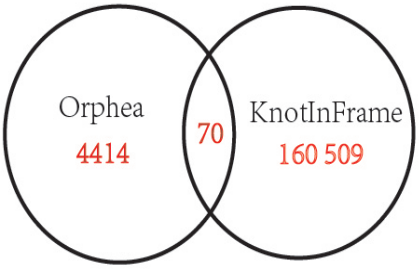
However, regarding the 1679 strong candidates of PRFdb (available in the Supplementary File *Frameshifting*), since the unavailability of the version of the *Saccharomyces cerevisiae* genome that was used in [Jacobs et al., 2007], I could



(a) Based on the *Saccharomyces cerevisiae* genome



(b) Based on the synthetic genome



(c) Based on 42 433 human mRNAs

Figure 4.8: The common prediction between Orphea and KnotInFrame.

Table 4.2: The prediction of Orphea and KnotInFrame based on three datasets.

	Size	Orphea	KnotInFrame	Common predictions
S.cerevisiae	> twelve million nucleotides	171	10118	4
Synthetic	twelve million nucleotides	102	9974	1
Human mRNAs	42 433 sequences	4414	160 509	70

not compare the PRFdb data with the predictions of Orphea and KnotInFrame based on a different version of *Saccharomyces cerevisiae* genome.

Indeed, KnotInFrame have stated that they have 74 common predictions with these 1679 strong candidates of PRFdb [Theis et al., 2008]. So I ran Orphea on the 1679 strong candidates of PRFdb directly to try to find the possibility of something interesting among Orphea and KnotInFrame and the strong candidates of PRFdb.

The result was that 12 matches were found between Orphea and PRFdb, but the three approaches did not have any common agreement, as shown in Figure 4.9. Additionally, there is not so much sense to compare the predictions between Or-

phea and KnotInFrame based on the 1679 strong candidates, as the unavailability of the entire genome and consequently the exhaustive predictions of Orphea and KnotInFrame. This corresponds to an empty subset in the intersection between Orphea and KnotInFrame in Figure 4.9.

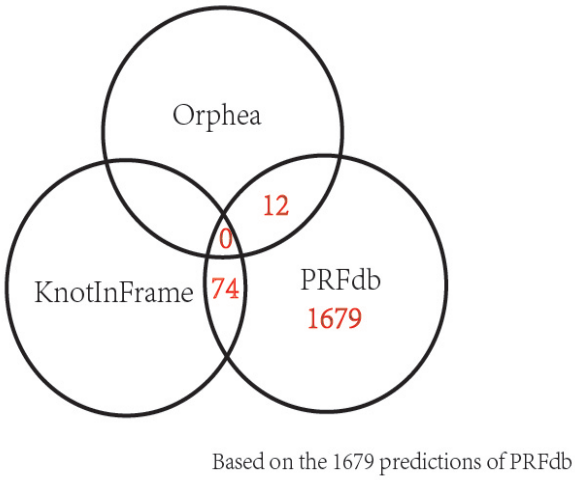


Figure 4.9: The common prediction among Orphea, KnotInFrame and PRFdb.

Discussion

After the ranking and selecting procedures mentioned in the Section 4.2.4, there were 49 predictions of Orphea chosen to be tested experimentally to promote a frameshifting *in vivo*, based on the *Saccharomyces cerevisiae* genome, the synthetic genome, and human mRNAs.

However, according to [Brégeon et al.], none of the common matches between Orphea and KnotInFrame, as shown in Figure 4.8, was among the 49 best predictions of Orphea which were tested empirically. Thus, we may have the conclusion that the best candidates of Orphea obtaining a strong frameshifting rate in the biological experiments had not been detected by KnotInFrame.

And as the reason of the unknown version of the *Saccharomyces cerevisiae* genome used by PRFdb, there is no obvious hints obtained from the comparison based on the 1679 strong candidates of PRFdb.

4.4.3 Comparison Based on the Best Predictions of Orphea

Dataset

As mentioned in Section 4.4.2, 49 predictions of Orphea were chosen to be tested experimentally, based on the *Saccharomyces cerevisiae* genome, the synthetic genome, and human mRNAs.

Learned from [Bekaert et al., 2003], the predictions having a frameshifting rate above 5% are considered as good predictions in this dissertation as well, and those having a frameshifting rate below 2% are considered as bad ones. Particularly, this part of comparison focuses on the good predictions, which include a collection of 6 best predictions of Orphea. They are listed in the descending order of their frameshifting rates in Table 4.3.

Some notations in the table are introduced here. For convenience, *Saccharomyces cerevisiae* genome is referred to as *Yeast*, the synthetic genome is referred to as *Random*, and human mRNAs is referred to as *Human*. Meanwhile, Orphea had 171 predictions based on the *Yeast* genome, 102 predictions based on the *Random* genome, and 4414 ones based on the *Human* genome respectively.

Aiming at distinguish the predictions, Orphea has assigned a reference number to each of its predictions, according to their detected positions in the input sequence. In this context, the *Sequence Name* in the first column of Table 4.3 are written in the order of *respective reference number in the predictions of Orphea_genome_frameshifting rate discovered in vivo*. For example, the 54_Random_0.179 represents the 54th prediction of Orphea, based on the synthetic genome with the frameshifting rate of 17.9% observed. Typically, their individual slippery positions are not shown in this nomenclature.

Methods

The comparison was carried out to test whether some other state-of-the-art programs can agree with the 6 best predictions of Orphea. The programs used were *CyloFold* [Bindewald et al., 2010], *IPknot* [Sato et al., 2011], *pknotsRG* [Reeder et al., 2007], *DotKnot* [Sperschneider and Datta, 2010], and *Vsfold5* [Dawson et al., 2007]. All of them were run through web services.

Particularly, the programs of IPknot, pknotsRG and DotKnot are labeled with several trailing letters in Table 4.3, corresponding to several variant algorithms that are adopted by each program to calculate the energy and fold the pseudoknot. In detail, IPknot-2 denotes that IPknot predicts the pseudoknot with the number of decomposed levels of 2, IPknot-3 denotes that IPknot predicts the structures with the number of decomposed levels of 3 [Sato et al., 2011]. pknotsRG-M and pknotsRG-F represent the standard **M**FE folding and en**F**orced folding algorithms of pknotsRG respectively [Reeder et al., 2007]. DotKnot and DotKnot-K represent the detection of standard **P**seudoknotted folding and the ones preferring the conformations of **K**issing hairpin [Sperschneider and Datta, 2010].

Results

The results shown in Table 4.3 were obtained by feeding the programs directly with the 6 best predictions of Orphea.

Inside the table, a '*yes*' means that the corresponding program can predict a pseudoknot in the given sequence, otherwise a 'no'.

The rightmost column *Score* of Table 4.3 shows the number of programs which detect a pseudoknot in the given sequence. And the nethermost rows *Overall* concludes the overall number of pseudoknots predicted by each program, based on the 6 best predictions of Orphea.

The precise comparison of the 6 best predictions of Orphea with the corresponding ones predicted by the other programs are provided in Appendix A, as well as the comparison that is based on the left 43 empirically tested structures of Orphea.

Table 4.3: The general comparison of 6 best predictions of Orphea.

Sequence Name	**KnotIn Frame**	**CyloFold**	**IPknot-2**	**IPknot-3**	**pknotsRG-M**	**pknotsRG-F**	**DotKnot**	**DotKnot-K**	**MCFold**	**Vsfold5**	**Scores (YES/10)**
54_Random _0.179	no	yes	yes	yes	yes	yes	yes	yes	no	no	7/10
3406_Human _0.1332	yes	no	no	no	no	yes	no	no		no	2/10
57_Random _0.131	no	yes	yes	yes	yes	yes	yes	yes	yes	no	8/10
4335_Human _0.0881	yes	yes	no	no	yes	yes	yes	yes	no	yes	7/10
1679_Human _0.0592	no	yes	no	no	no	yes	yes	yes	yes	yes	6/10
4339_Human _0.0558	yes	yes	no	no	yes	yes	yes	yes	yes	no	7/10
Overall	3/6	5/6	2/6	2/6	4/6	6/6	5/6	5/6	3/6	2/6	

Discussion

According to Table 4.3, the programs have responded diversely to the 6 best predictions of Orphea. We cannot have a prominent conclusion of the most 'popular' sequences clearly. It seemed that the best predictions of Orphea with efficient frameshifting rates were not identified well by all the programs, especially the 3406_Human_0.1332.

But generally, CyloFold and DotKnot had a relatively good performance, as they detected 5 sequences out of 6. Especially, pknotsRG has obtained a very good sensitivity (100%) on predicting pseudoknots with its Enforced folding algorithm as it predicted all the predictions, but offering no guarantee on the quality of the prediction. The enforcing pseudoknotted folding algorithm focuses on searching a pseudoknot globally, in spite of the free energy of the folded structure. Contrarily, pknotsRG with the MFE algorithm focuses on finding a more stable structure with the lowest free energy, which may not contain pseudoknots.

Meanwhile, Table 4.3 shows that it seems difficult for a majority of methods to agree with the best predictions of Orphea, in spite of their high level of -1 PRF rates obtained *in vivo*. This may be because of the diversity of the calculating algorithms and the predicting parameters and models which are restrained greatly by the available structures in the database to learn and the functional and mechanical knowledge of pseudoknots.

Particularly, this round of comparisons can not be assessed by the evaluation parameters as introduced in Section 4.3.1, as the 6 best predictions of Orphea can not be referred to as the reference structures.

4.4.4 Comparison Based on the Frameshifting Signals in PseudoBase

This section is about the comparison between the prediction of Orphea and KnotInFrame on the reference structures of the frameshifting signals in PseudoBase.

Datasets

There are 34 viral frameshifting signals in PseudoBase [Van Batenburg et al., 2000], as of January 22, 2015.

Particularly, 17 signals were chosen as the learning data of Orphea [Brégeon et al.; Forest, 2005], as mentioned in Section 4.2.4. Consequently, the comparison was carried out in two rounds. The first one was based on the 17 learning sequences of Orphea. And the second one was based on the other 17 frameshifting signals in PseudoBase, which are referred to as the *testing data* of Orphea below.

The information of the 17 learning frameshifting signals of Orphea are shown in Table 4.4 in detail, including the reference number in PseudoBase, with the prefix of *PKBNo.*, the organisms they come from, and the corresponding sizes of the submotifs of a frameshifting signal. Table 4.5 shows the similar information of the 16 testing frameshifting signals of Orphea, where one signal is excluded as its stimulating pseudoknot is a kissing hairpin. Particularly, it is the *Human Coronavirus 229E (HCV_229E)*, with the reference number of PKB171 and a length of 224 nucleotides.

Table 4.6 shows the information of the *Human Coronavirus 229E (HCV_229E)* provided in PseudoBase, where ':::' represents the unpaired region in the reference structure. This unpaired region corresponds to the horizontal unpaired line between two hairpins in Figure 4.10, which is drawn with all 224 nucleotides. We supposed a free unpaired region between the two hairpins, because of the omitted secondary information in PseudoBase.

Table 4.4: The 17 learning frameshifting signals of Orphea in PseudoBase.

Gene Name	PKBNo.	Organism	Length (nt)	Spacer (nt)	Stem1 (bp)	Stem2 (bp)	Loop1 (nt)	Loop2 (nt)	Loop3 (nt)
BLV	PKB1	Bovine Leukemia Virus	34	7	6	3	5	0	4
BWYV	PKB2	Beet Western-Yellows Virus	32	6	5	4	2	0	6
EIAV	PKB3	Equine Infectious Anemic Virus	44	9	6	4	3	0	12
FIV	PKB4	Feline Immunodeficiency Virus	43	8	5	6	2	0	11
PLRV-W	PKB42	Potato Leafroll Virus	32	6	4	3	2	1	9
PLRV-S	PKB43	Potato Leafroll Virus	32	6	4	4	2	0	8
CABYV	PKB44	Cucurbit Aphid-Borne Yellows Virus	32	5	5	3	2	1	8
PEMV	PKB45	Pea Enation Mosaic Virus	34	6	6	4	2	0	6
BYDV-NY-RPV	PKB46	Barley Yellow Dwarf Virus	32	5	5	4	2	0	7
MMTV_gag/pro	PKB80	Mouse Mammary Tumor Virus	41	7	5	7	1	1	8
IBV	PKB106	Infectious Bronchitis Virus	75	6	11	7	1	0	32
SRV1_gag/pro	PKB107	Simian RetroVirus-1	44	7	6	6	1	0	12
BEV	PKB128	Berne Virus	110	5	11	5	4	0	69
LDV-C	PKB217	Lactate Dehydrogenase-elevating Virus	65	6	11	6	3	3	19
PRRSV-16244B	PKB218	Porcine Reproductive Respiratory Syndrome Virus	63	5	12	7	4	0	15
PRRSV-LV	PKB233	Porcine Reproductive Respiratory Syndrome Virus	63	5	12	7	4	0	15
BChV	PKB240	Beet Chlorosis Virus	33	7	4	4	1	1	8

Table 4.5: The 16 testing frameshifting signals of Orphea in PseudoBase.

Gene Name	PKBNo.	Organism	Length(nt)	Spacer(nt)	Stem1(bp)	Stem2(bp)	Loop1(nt)	Loop2(nt)	Loop3(nt)
EAV	PKB127	Equine Arteritis Virus	122	6	11	6	2	1	69
RSV	PKB174	Rous Sarcoma Virus	128	1	14	8	11	52	11
WBV	PKB253	White Bream Virus	82	4	14	5	4	0	29
SARS-CoV	PKB254	SARS Coronavirus	82	5	11	7	3	0	29
Mm_Edr	PKB257	Mus Musculus (mouse)	66	5	10	9	3	0	9
Hs_Ma3	PKB258	Homo Sapiens	60	6	11	5	3	1	10
VMV	PKB280	Visna-Maedi Virus	68	6	7	7	5	7	14
ScYLV	PKB281	Sugarcane Yellow Leaf Virus	43	6	5	3	2	1	9
KUNV	PKB346	West Nile virus, Kunijn Subtype	75	5	11	7	6	3	17
WNV	PKB347	West Nile virus	75	5	11	7	6	3	17
JEV	PKB348	Japanese Encephalitis Virus	77	5	11	7	7	2	16
MVEV	PKB349	Murray Valley Encephalitis Virus	80	5	11	7	7	2	16
ALFV	PKB350	Alfuy Virus	77	5	11	6	8	2	16
USUV	PKB351	Usutu Virus	80	5	11	7	7	2	16
MIDV	PKB352	Middelburg Virus	70	6	10	7	3	1	13
SESV	PKB353	Seal Louse Virus	70	7	11	8	1	0	9

Table 4.6: The sequence and secondary structure of the *Human Coronavirus 229E (HCV_229E)*, PKB171 in PseudoBase.

Human Coronavirus 229E (HCV_229E): PKB171, 224 nucleotides	
Sequence	UUUAAACGAGUCCGGGGCUCUAGUGCCGCUCGACUAGAGCCCUGUAA:::CAGUUAUGGACCACGAGCAGUCCAUGUA
Structure	............(((((((((((((...[[[[[)))))))))))))...:::::....(((((((..]]]]].)))))))..

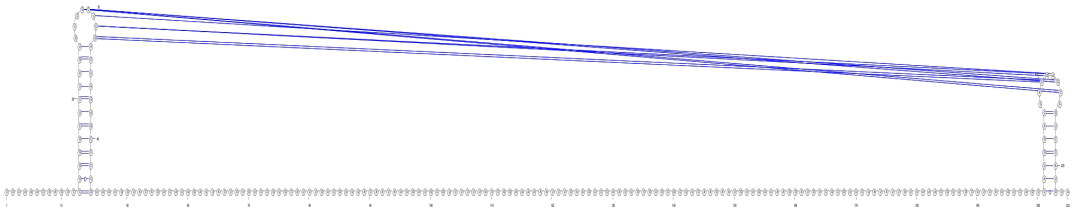


Figure 4.10: The reference structure of the *Human Coronavirus 229E (HCV_229E)*.

Results

The comparison of the predicted structures with the reference structures in PseudoBase are partially shown in Table B.2. The entire comparison can be found in the Appendix B.

Quite remarkably, Orphea failed twice to predict a pseudoknot based on the 17 learning signals, and failed six times based on the 17 testing signals. On the other hand, KnotInFrame failed eight times based on the 17 learning signals, and failed five times based on the 17 testing signals.

Table 4.8 and Table 4.9 show the performance of the corresponding predictions by Orphea and KnotInFrame based on the 17 learning frameshifting signals of Orphea, calculated by Equations 4.1, 4.2 and 4.3. Table 4.10 and Table 4.11 show the performance of corresponding predictions by Orphea and KnotInFrame based on the 17 testing frameshifting signals of Orphea.

Table 4.7: Three examples of the comparison with the reference structures in PseudoBase.

Gene Name	**Program**	**Result**
LDV-C (PKB217)	Sequence	UUUAAACUGCUAGCCACCUCUGGUCUCGACCGCUGUACUAGAGGUGGGCUGACGGUGUYUGGCGAUGCGGUCA
	Slippery Site	UUUAAAC
	SubSequence	UGCUAGCCACCUCUGGUCUCGACCGCUGUACUAGAGGUGGGCUGACGGUGUYUGGCGAUGCGGUCA
	PseudoBase	......((((((((((...[[[[[[...))))))))))..................]]]]]].
	Orphea	......((((((((((.....[[[[[[))))))))))..........]]]]]]..........
	KnotInFrame	......((((((((((.....[[[[[[))))))))))....]]]]]]...............
BEV (PKB128)	Sequence	UUUAAACUGUUGAGAGGUGCCUGGAGCGCCUGCAGGCAUCUCUGUUUUCAAAAUGGCGCAUACCAGUCUUCAAGGUCAAAACAUUAUAUUGAUUUGGCAACUGAGUAUAAUGCAGGCA
	Slippery Site	UUUAAAC
	SubSequence	UGUUGAGAGGUGCCUGGAGCGCCUGCAGGCAUCUCUGUUUUCAAAAUGGCGCAUACCAGUCUUCAAGGUCAAAACAUUAUAUUGAUUUGGCAACUGAGUAUAAUGCAGGCA
	PseudoBase	.....((((((((((....[[[[[))))))))))..... ....]]]]].
	Orphea	.....((((((((((..[[[[[..))))))))))..........]]]]]..... .....
	KnotInFrame	.....((((((((((..[[[[[..))))))))))..........]]]]]..... .....

Continued On Next Page

Table 4.7 – *Continued From Previous Page*

Gene Name	Program	Result
HCV_ 229E (PKB171)	Sequence	UUUAAACGAGUCCGGGGCUCUAGUGCCGCUCGACUAGAGCCCUGUAAUGGUACAGACAUAGAUUACUGUGUCCGUGCAUUUGACGUUUACAAU AAAGAUGCGUCUUUUAUCGGAAAAAAUCUGAAGUCCAAUUGUGUGCGCUUCAAGAAUGUAGAUAAGGAUGACGCGUUCUAUAUUGUUAAACGU UGCAUUAAGUCAGUUAUGGACCACGAGCAGUCCAUGUA
	Slippery Site	UUUAAAC
	SubSequence	GAGUCCGGGGCUCUAGUGCCGCUCGACUAGAGCCCUGUAAUGGUACAGACAUAGAUUACUGUGUCCGUGCAUUUGACGUUUACAAUAAAGAUG CGUCUUUUAUCGGAAAAAAUCUGAAGUCCAAUUGUGUGCGCUUCAAGAAUGUAGAUAAGGAUGACGCGUUCUAUAUUGUUAAACGUUGCAUUA AGUCAGUUAUGGACCACGAGCAGUCCAUGUA
	PseudoBase	.....((((((((((((...[[[[[)))))))))))))..... ....(((((((..]]]]].)))))))..
	Orphea	.....((((((((((((.[[[....))))))))))))...]]]..... ....
	KnotInFrame	.....((((((((((((.[[[....))))))))))))).....((((.(((((((....)))))))).)))).....(((((.....)))))......]]]..... .....

Tables 4.12 and 4.13 conclude the global performance of Orphea and KnotInFrame in predicting the 34 frameshifting signals, where the higher values based on each frameshifing signal are highlighted in bold.

Quick remarkably, Orphea prevailed over KnotInFrame globally.

Based on the learning data, Orphea successfully detected more pseudoknots than KnotInFrame, and obtained an average higher values of sensitivity, PPV and MCC than KnotInFrame. Based on the testing data, Orphea won for its higher values as well, in spite of failing one more time in detecting pseudoknot than KnotInFrame. Particularly, Orphea has obtained 11 precise predictions compared to the reference structures, with their MCC values equal to 1. On the other hand, KnotInFrame has obtained several negative MCC values, which shows the disagreement between the predictions and the reference structures in different levels [Matthews, 1975].

Table 4.8: The 15 predictions of Orphea based on 17 learning signals.

Name	PKBNo.	TP	TN	FP	FN	Sensitivity	PPV	MCC
BLV	PKB1	8	217	0	1	0.889	1.0	0.941
BWYV	PKB2	9	253	0	0	1.0	1.0	1.0
EIAV	PKB3	10	306	0	0	1.0	1.0	1.0
FIV	PKB4	11	269	0	0	1.0	1.0	1.0
PLRV-S	PKB43	8	105	0	0	1.0	1.0	1.0
CABYV	PKB44	2	124	6	6	0.25	0.25	0.204
BYDV-NY-RPV	PKB46	3	137	6	6	0.333	0.333	0.291
MMTV_gag-pro	PKB80	11	288	0	1	0.917	1.0	0.956
IBV	PKB106	17	1143	0	1	0.944	1.0	0.971
SRV1_gag-pro	PKB107	12	307	0	0	1.0	1.0	1.0
BEV	PKB128	11	2313	5	5	0.688	0.688	0.685
LDV-C	PKB217	11	827	6	6	0.647	0.647	0.64
PRRSV-16244B	PKB218	19	822	1	0	1.0	0.95	0.974
PRRSV-LV	PKB233	19	783	0	0	1.0	1.0	1.0
BChV	PKB240	3	131	4	5	0.375	0.429	0.368

Table 4.9: The 9 predictions of KnotInFrame based on 17 learning signals.

Name	PKBNo.	TP	TN	FP	FN	Sensitivity	PPV	MCC
BLV	PKB1	6	216	3	3	0.667	0.667	0.653
BWYV	PKB2	9	253	0	0	1.0	1.0	1.0
MMTV_gag-pro	PKB80	5	289	5	7	0.417	0.5	0.436
IBV	PKB106	0	1149	11	18	0.0	0.0	-0.012
SRV1_gag-pro	PKB107	5	311	3	7	0.417	0.625	0.495
BEV	PKB128	11	2313	5	5	0.688	0.688	0.685
LDV-C	PKB217	11	827	6	6	0.647	0.647	0.64
PRRSV-16244B	PKB218	6	827	12	13	0.316	0.333	0.31
PRRSV-LV	PKB233	0	785	17	19	0.0	0.0	-0.022

Table 4.10: The 11 predictions of Orphea based on 17 testing signals.

Name	PKBNo.	TP	TN	FP	FN	Sensitivity	PPV	MCC
EAV	PKB127	11	2522	3	6	0.647	0.786	0.711
HCV_229E	PKB171	12	9098	3	12	0.5	0.8	0.632
WBV	PKB253	16	1062	0	3	0.842	1.0	0.916
SARS-CoV	PKB254	14	1082	0	12	0.538	1.0	0.73
Hs_Ma3	PKB258	16	431	0	0	1.0	1.0	1.0
VMV	PKB280	14	617	0	0	1.0	1.0	1.0
ScYLV	PKB281	8	135	1	0	1.0	0.889	0.939
WNV	PKB347	18	877	0	2	0.9	1.0	0.948
JEV	PKB348	18	939	0	2	0.9	1.0	0.948
ALFV	PKB350	16	922	0	1	0.941	1.0	0.97
SESV	PKB353	19	655	0	0	1.0	1.0	1.0

Table 4.11: The 12 predictions of KnotInFrame based on 17 testing signals.

Name	PKBNo.	TP	TN	FP	FN	Sensitivity	PPV	MCC
HCV_229E	PKB171	12	9098	19	12	0.5	0.387	0.438
RSV	PKB174	32	2650	3	7	0.821	0.914	0.864
WBV	PKB253	14	1060	4	5	0.737	0.778	0.753
SARS-CoV	PKB254	10	1082	4	16	0.385	0.714	0.516
Mm_Edr	PKB257	5	552	4	14	0.263	0.556	0.369
Hs_Ma3	PKB258	0	438	9	16	0.0	0.0	-0.027
VMV	PKB280	13	618	3	1	0.929	0.813	0.865
WNV	PKB347	11	878	6	9	0.55	0.647	0.588
JEV	PKB348	11	940	6	9	0.55	0.647	0.589
ALFV	PKB350	7	925	6	10	0.412	0.538	0.463
MIDV	PKB352	14	607	3	3	0.824	0.824	0.819
SESV	PKB353	11	659	4	8	0.579	0.733	0.643

Table 4.12: The comparison of predictions of Orphea and KnotInFrame based on 17 learning signals.

Name	PKBNo.	Sensitivity		PPV		MCC	
		Orphea	KIF	Orphea	KIF	Orphea	KIF
BLV	PKB1	**0.889**	0.667	**1.0**	0.667	**0.941**	0.653
BWYV	PKB2	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
EIAV	PKB3	**1.0**		**1.0**		**1.0**	
FIV	PKB4	**1.0**		**1.0**		**1.0**	
PLRV-S	PKB43	**1.0**		**1.0**		**1.0**	
CABYV	PKB44	**0.25**		**0.25**		**0.204**	
BYDV-NY-RPV	PKB46	**0.333**		**0.333**		**0.291**	
MMTV_gag-pro	PKB80	**0.917**	0.417	**1.0**	0.5	**0.956**	0.436
IBV	PKB106	**0.944**	0.0	**1.0**	0.0	**0.971**	-0.012
SRV1_gag-pro	PKB107	**1.0**	0.417	**1.0**	0.625	**1.0**	0.495
BEV	PKB128	**0.688**	**0.688**	**0.688**	**0.688**	**0.685**	**0.685**
LDV-C	PKB217	**0.647**	**0.647**	**0.647**	**0.647**	**0.64**	**0.64**
PRRSV-16244B	PKB218	**1.0**	0.316	**0.95**	0.333	**0.974**	0.31
PRRSV-LV	PKB233	**1.0**	0.0	**1.0**	0.0	**1.0**	-0.022
BChV	PKB240	**0.375**		**0.429**		**0.368**	
Overall		**0.803**	0.461	**0.82**	0.496	**0.802**	0.465

Table 4.13: The comparison of predictions of Orphea and KnotInFrame based on 17 testing signals.

Name	PKBNo.	Sensitivity		PPV		MCC	
		Orphea	KIF	Orphea	KIF	Orphea	KIF
EAV	PKB127	**0.647**		**0.786**		**0.711**	
HCV_229E	PKB171	**0.5**	**0.5**	**0.8**	0.387	**0.632**	0.438
RSV	PKB174		**0.821**		**0.914**		**0.864**
WBV	PKB253	**0.842**	0.737	**1.0**	0.778	**0.916**	0.753
SARS-CoV	PKB254	**0.538**	0.385	**1.0**	0.714	**0.73**	0.516
Mm_Edr	PKB257		**0.263**		**0.556**		**0.369**
Hs_Ma3	PKB258	**1.0**	0.0	**1.0**	0.0	**1.0**	-0.027
VMV	PKB280	**1.0**	0.929	**1.0**	**0.813**	**1.0**	0.865
ScYLV	PKB281	**1.0**		**0.889**		**0.939**	
WNV	PKB347	**0.9**	0.55	**1.0**	0.647	**0.948**	0.588
JEV	PKB348	**0.9**	0.55	**1.0**	0.647	**0.948**	0.589
ALFV	PKB350	**0.941**	0.412	**1.0**	0.538	**0.97**	0.463
MIDV	PKB352		**0.824**		**0.824**		**0.819**
SESV	PKB353	**1.0**	0.579	**1.0**	0.733	**1.0**	0.643
Overall		**0.843**	0.546	**0.952**	0.629	**0.89**	0.573

Discussion

According to Tables 4.12 and 4.13, Orphea and KnotInFrame both failed to predict some frameshifting signals some times.

The reason to explain the failures of Orphea may be that the input sequences are basically too small. This results in either the threshold score assigned during the searching phase is not reached for the ES1, or the length is too small to find ES2 when a valid ES1 is found.

The reason that KnotInFrame fails to predict a structure may be because the length of some sequences are too short to reach the threshold of the input requirement. However, KnotInFrame can occasionally have a not bad prediction, compared to the reference structure, once they are elongated with an AAA-tail, to the length of around 50 nucleotides. Another prominent explanation of the failures of KnotInFrame is that KnotInFrame requires a much more restricted slippery sequence, namely three identical nucleotides for X in $X\,XXY\,YYZ$.

And again, according to Tables 4.12 and 4.12, it is clear to have the following conclusions. First, Orphea had an overwhelming triumph as it obtained higher values of sensitivity, PPV and MCC, based on most reference structures. Second, Orphea is more sensitive, especially to short sequences than KnotInFrame, and can tolerate a more general composition of slippery sequence in predicting -1 PRF signals.

On the other hand, for the genes of BWYV, BEV, LDV-C and HCV_229E, Orphea has obtained a draw with KnotInFrame for their equal values of sensitivity, PPV and MCC. However, it is very interesting to reveal that Orphea has predicted a different pseudoknot from KnotInFrame although both programs have identical evaluation values. The examples are LDV-C and HCV_229E, as shown in Table B.2. This further explains the dilemma we mention in Section 4.3 where we notice the unsatisfactory assessments by sensitivity, PPV and MCC as calculated by Equations 4.1, 4.2 and 4.3, but there are no better options of evaluation parameters for us to choose and no reliable suggestions on a practical applicability of them to benefit.

4.5 Conclusion

This chapter has introduced one typical recoding event, -1 programmed ribosomal frameshifting, where the downstream pseudoknot plays the role of a strong stimulator. Brief introduction of four programs predicting -1 PRF signals, FS-Finder, the corresponding work of PRFdb, KnotInFrame and Orphea followed.

This chapter principally focused on a series of comparisons of predicting -1 programmed ribosomal frameshifting signals by the available methods. For the assessment of the predictions, three evaluation parameters and their variants were discussed as well.

However, the pseudoknots involved in frameshifting are only a subset of the pseudoknots family, many other pseudoknots may take care of the other types of recoding events and much more general molecular procedures. So in the following chapters, the study of more general pseudoknots is going to be introduced, including the classification of pseudoknots, and prediction of pseudoknots by the

contemporary methods.

Typically, we term this coming work as a benchmark for the pseudoknots and the prediction methods.

Chapter 5

Preparation of the Benchmark for Pseudoknots and Prediction Methods

This chapter opens a new topic of my work, a benchmark particularly designed for the RNA pseudoknots and the prediction methods. As the main contributions of this dissertation, this part of work is organized in three chapters.

This chapter introduces the motivation and preliminaries of this benchmark, as well as the preparation work, such as the datasets and methods involved, the characteristics of pseudoknots considered, and the evaluation parameters employed.

Chapter 6 shows the results obtained, including both the classification of pseudoknots, and the prediction of pseudoknots. Further, the benchmark is accessible with an on-line version for the community, suggesting some details of the web development.

Based on the results shown in Chapter 6, the respective discussions are aroused in Chapter 7, and the conclusion about the benefits and lessons we may obtain from this benchmark.

5.1 Motivation

Comparison of the predicting results from more than one program is a good approach to generating an informed hypothesis about RNA structure and func-

tion [Schroeder, 2009]. The evaluation criteria and databases of known secondary structures used to evaluate prediction accuracy vary substantially between different research groups and make direct comparisons complex [Schroeder, 2009]. Comparing, or benchmarking the predictions returned by different RNA structure prediction methods, which are based on the sequences in the same dataset, suggests an acceptable and reliable comparison.

However, such kind of benchmarking systems for assessing the RNA secondary structure prediction methods is rarely performed, compared to other bioinformatic domains, such as the practice of the protein-folding algorithms [Eyrich et al., 2001], protein-protein docking [Chen et al., 2003] and multiple sequence alignment [Gardner et al., 2005; Thompson et al., 1999]. This is significant to bear in mind which of the available methods for secondary structure prediction is the most accurate and useful practically, as an increasing number of efforts have been made on the exploration of more powerful *in silico* prediction methods.

As one of the pioneers, *BRAliBase I* by [Gardner and Giegerich, 2004] benchmarks the comparative RNA structure prediction algorithms, which are preferred as the homologous RNA sequences are available. But BRAliBase I did neither pay much attentions on the RNA structure prediction methods based on a single sequence, nor on the pseudoknots.

Recently, [Puton et al., 2013] proposes a benchmark of RNA structure prediction methods, *CompaRNA*. CompaRNA focuses on the methods predicting an RNA secondary structure either from a single sequence or from the comparative analysis when a set of homologous sequences are available. And CompaRNA considers both the pseudoknot-free secondary structures and the pseudoknots concurrently.

More precisely, CompaRNA considers pairs of programs at a time exhaustively, and compares the mean evaluation values based on the dataset to which both programs return a secondary structure. The programs are ranked according to the number of being a winner in the pair-wise comparisons.

However, CompaRNA does not separate the comparative structure prediction methods and the ones based on a single sequence, and compares them equally on the benchmarking datasets. This suggests few insights into the predictions

of the 'single-sequence' methods, which is caused by their generally worse performance compared to the comparative methods, as the latter holds more information from the set of homologous sequences provided. This is further supported by the ranking result obtained by CompaRNA on the PDB dataset (Table 5 therein) where comparative methods have obtained an overwhelming dominance [Puton et al., 2013]. On the other hand, the datasets in CompaRNA correspond to particular collections of sequences which are determined by the pairs of methods in comparison, as each sequence is returned with a secondary structure by both programs. It implies that all the methods are not evaluated with the consistent set of sequences, suggesting CompaRNA's ranking system is not global enough. In addition, CompaRNA pays few attention on the failure of predictions in the pairwise comparisons. Last but not least, CompaRNA takes the ranking generated on pseudoknots as only a particular case.

This part of work is going to introduce a benchmark of RNA secondary structure prediction methods which particularly focuses on the ones predicting RNA pseudoknots from a given sequence. The primary purpose and contribution of this benchmark is to take advantage of the existing methods to generate a practical prediction for the given sequence. It is relative to the questions of how to carry out a reasonable prediction, how to make a proper selection of prediction methods for the given sequence, and how much accuracy the prediction holds.

Meanwhile, a good knowledge on the characteristics of pseudoknots will promote a more persuasive comparison of predictions. This arouses the second contribution of this benchmark, a comprehensive analysis of the pseudoknot classifications, according to several categories of complexity measures.

In practice, this benchmark considers a *common* or *shared* set of sequences where each sequence is returned with a secondary structure by all the benchmarking methods. Further, the predictions are assessed on hierarchical subsets of this common set which are divided by the length, organism and RNA type of the sequences, and the classifications that the implied pseudoknots are subject to. In addition, the sequences which are returned with a secondary structure by some methods and are not by some other methods are considered as an *uncommon* or *missing* set. The predictions based on this set of sequences are compared as well,

which is expected to reflect different levels of failure of prediction by the prediction methods.

5.2 Datasets

There are two provenances of pseudoknots used for benchmarking in this chapter, one provenance is database *PseudoBase* [Van Batenburg et al., 2000], a particular database for the pseudoknots, and the other is the set of some pseudoknotted entries from the database *Protein Data Bank (PDB)* [Berman et al., 2000]. More precisely, there are:

- 367 pseudoknots from PseudoBase, as of March 28, 2014. The sequences, PKB1-PKB367 have been downloaded from the database directly. Particularly, PseudoBase focuses on the crossing interactions forming the pseudoknot, and omits partial structural information elsewhere for 27 relatively long sequences, such as the PKB64 and PKB192. An example PKB171 is shown in Table 4.6, where the ':::' represents the unknown potential details in PseudoBase, and Figure 4.10. In this benchmark, the corresponding unknown parts are referred to as unpaired bases consistently, and the complete sequences and structures are provided in the Supplementary File *Benchmark*. Additionally, the first 304 pseudoknots in this dataset, PKB1-PKB304, correspond to the records in the *PseudoBase++* [Taufer et al., 2009].

- 47 pseudoknots extracted from PDB, which are provided by *CompaRNA*'s authors kindly, as of June 5, 2013. Specifically, CompaRNA uses several filters to select PDB records for benchmarking, such as the restrictions on the length which should be longer than 20 nucleotides, and the RNA backbone which should be continuous [Puton et al., 2013].

The pair-wise interactions of the total 414 pseudoknots contain only the canonical A-U, G-C pairs and the wobble G-U pair. The collection of such types of base pairs are also referred to as the *standard* base pairs. And we do not consider the non-canonical interactions [Leontis and Westhof, 2001].

Additionally, the *isolated base pair* is allowed in this benchmark, while the *triples* are forbidden. The notion of *isolated* base pair corresponds to a stem with only one base pair, and the *triples* are those bases which participate in the base pairing with two partners at the same time.

The details of the 414 sequences are provided in Supplementary File *Benchmark*.

5.3 Classification of Pseudoknots

The pseudoknots can be classified according to hierarchical complexity measures, such as the physical interactions, algorithmic accessibilities and conformational characteristics, which refer to the physical interaction of stems, the algorithms that can predict them, and some conformational characteristics respectively.

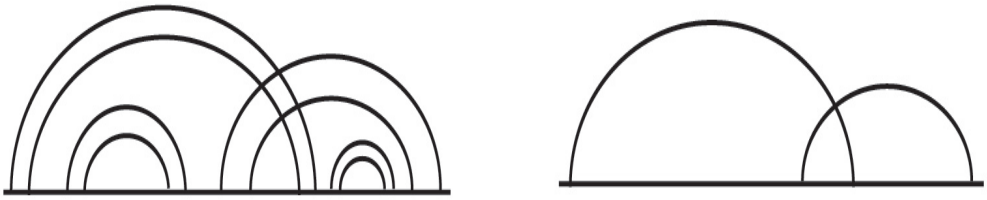
The pseudoknots can be sorted into classes by some other criteria, such as the mathematical definitions proposed in [Han et al., 2008; Wong et al., 2011]. But this dissertation has more interests on the first three classifications.

5.3.1 Physical Interactions

Pseudoknots are formed by non-nested base pairs. A general definition of the conformation of a pseudoknotted structure is that an unpaired loop region in a classical secondary structure is involved in the standard base-pairings with a complementary region outside this loop [Pleij, 1993].

As a preliminary, the *shadow* of a RNA secondary structure is obtained by removing all non-crossing arcs, collapsing all unpaired bases, and replacing all adjacent parallel arcs by single arcs, with the loss of some information on the size of the stems and non-crossing components of the global structure [Reidys et al., 2011]. A schematic figure is shown in Figure 5.1.

In the context of pseudoknot study, the RNA shadow captures the dominant interactions forming the pseudoknot, in spite of sacrificing some details. Consequently, this dissertation declares each pseudoknot a particular physical type against the corresponding RNA shadow.



(a) An H-type pseudoknot (b) The shadow

Figure 5.1: An H-type pseudoknot and its shadow.

As an extension of the introduction of the structures with pseudoknots in Chapter 2, this dissertation concludes the pseudoknots family as the following four types principally, from the aspect of physical interaction of the stems:

- The *H-type* pseudoknots, as described in Section 2.2.3. This most prevalent pseudoknot type covers the major members of PseudoBase with the pseudoknot pattern of *ABAB*. An example is the *gag/pro ribosomal frameshifting* pseudoknot of the *simian retrovirus-1 (SRV1_gag/pro)* with the reference number of PKB107 in PseudoBase.

- The *kissing hairpin* pseudoknots, or kissing hairpins for short, as introduced in Section 2.2.3. An example of kissing hairpin is the pseudoknot present in the *coxsackie B virus (CoxB3)*, with the pattern of *ABACBC* and reference number of PKB169 in PseudoBase.

- The *recursive* pseudoknots, where a pseudoknot is locally embedded in the unpaired single-strand region of another pseudoknot. The embedding and embedded pseudoknot can be either an H-type pseudoknot or a kissing hairpin. In fact, the beginning and ending loops of the pseudoknot hold the possibility to harbor a substructure locally as well, which makes the recursive pseudoknot a conformation of several consecutive pseudoknots. An example of this case is the pseudoknot found in the *Thermus thermophilus tmRNA*, with the pattern of *ABAB* and reference number of 3IYQ in PDB.

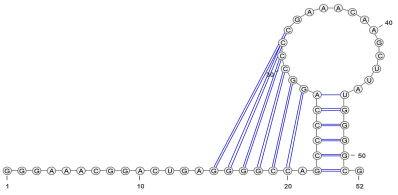
- The *complex* pseudoknots, which contain more complex interactions than

the previous three types. A common case is the *pseudotrefoil* pseudoknot present in the *Escherichia coli (Ec_alpha)* α mRNA, with the pattern of *ABCABC* and reference number of PKB71 in PseudoBase. Another example is the pseudoknot found in the ribozyme of the *Hepatitis delta virus*, with the pattern of *ABCDCADB* and reference number of PKB75 in PseudoBase.

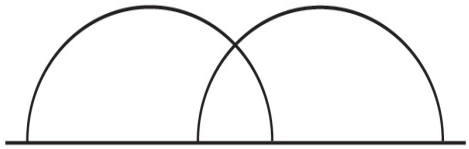
Figure 5.2 shows these four types of pseudoknots, where subfigures 5.2(a), 5.2(c), 5.2(e) and 5.2(g) are the visualizations of PKB107, PKB169, 3IYQ and PKB71 by VARNA, and the 5.2(b), 5.2(d), 5.2(f) and 5.2(h) are their corresponding RNA structure shadows. Particularly, there are four consecutive, or 'independent' H-type pseudoknots in 3IYQ, as shown in Figures 5.2(e) and 5.2(f). We prefer to declare its pseudoknot pattern of *ABAB* as the global shadow in the following chapters, rather than *ABABCDCDEFEFGHGH* as they are four identical H-type pseudoknots.

As the variants of H-type pseudoknots, [Pleij, 1993] introduces the *bulge-type (B-type)* pseudoknot and the *interior-type (I-type)* pseudoknot, which have the same pseudoknot pattern of *ABAB*. Specifically, instead of the hairpin loop, the unpaired nucleotides in a bulge loop or an interior loop can base pair with a region outside the loop, constructing the rare B-type pseudoknot and I-type pseudoknot. Similarly, once the unpaired nucleotides in the multi-loop are involved in forming a pseudoknot, the pseudoknot can be classified as the *multi-loop-type (M-type)* pseudoknot.

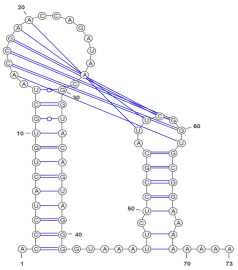
A typical example of the B-type pseudoknot is found in the tRNA-like structure at the end of the *tobacco mosaic virus (TMV)* [Pleij, 1993], with the reference number of PKB57 in PseudoBase. A typical example of the I-type pseudoknot is found in the *internal ribosomal entry site (IRES)* region in the *Plautia stali intestine virus (PSIV_IRES-PKIII)*, with the reference number of PKB212 in PseudoBase. An example of the M-type pseudoknot is found in the *viral frameshifting* pseudoknot of the *rous sarcoma virus (RSV)* with the reference number of PKB174 in PseudoBase.



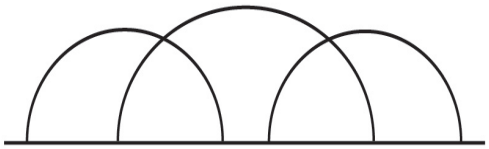
(a) *PKB107* in PseudoBase, an H-type pseudoknot.



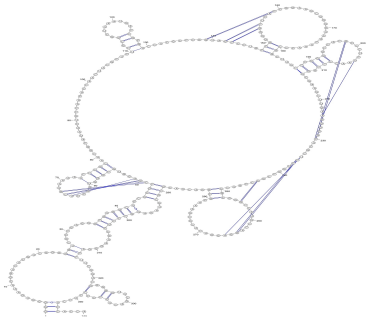
(b) The shadow of PKB107.



(c) *PKB169* in PseudoBase, a kissing hairpin.

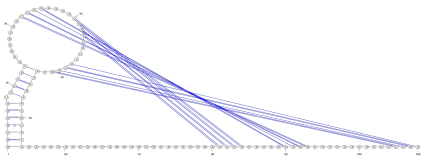


(d) The shadow of PKB169.

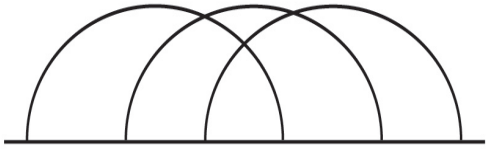


(e) *3IYQ* in PDB, a recursive pseudoknot.

(f) The shadow of 3IYQ, with the pseudoknot pattern of *ABAB* as the identical pseudoknots.



(g) *PKB71* in PseudoBase, a pseudotrefoil.



(h) The shadow of PKB71.

Figure 5.2: Physical classification of the pseudoknots.

5.3.2 Algorithmic Accessibilities

As mentioned in Section 3.2.1, the methods which are available to predict pseudoknots have more or less trade-offs between the practical consideration of the generality of pseudoknots and reasonable computer cost. Comparison of their performance of predictions will be more persuasive with a good knowledge on the characteristics of pseudoknots detected by them. In this algorithmic classification, the classes of pseudoknots are defined according to the specification of algorithms which can predict them or not, as done by [Condon et al., 2004] and [Saule et al., 2011]. Both researches contribute to the formal definitions of each algorithmic class of pseudoknots, and the inclusion relationships between them.

Lyngso & Pederson (L&P) Class

The set of pseudoknots that the *L&P*'s algorithm [Lyngsø and Pedersen, 2000a] can detect composes the *L&P class* of pseudoknots.

The basic component of the *L&P class* of pseudoknots is one H-type pseudoknot. Although the authors of the *L&P*'s algorithm report the possibility of containing a *B-type* or an *I type* pseudoknot in their model, the *L&P class* of pseudoknots is always referred to as the set of any number of pseudoknot free structures, and the structures with only one H-type pseudoknot.

Dirks & Pierce (D&P) Class

The set of pseudoknots that the *D&P*'s algorithm [Dirks and Pierce, 2003] can detect composes the *D&P class* of pseudoknots.

The basic component of the *D&P class* of pseudoknots is also one H-type pseudoknot, where two pseudoknot-free structures cross each other. But compared to the *L&P class*, the *D&P class* allows the recursion of any H-type pseudoknots arbitrarily concatenated and embedded inside the unpaired intervals, as shown in Figure 5.3(b), where the recursive regions are marked with *R*.

As a result, the set of the *D&P class* of pseudoknots consists of any number of pseudoknot free structures, H-type pseudoknots and their arbitrary concatenation and embedment inside each other, such as the pseudoknots with the patterns of

ABCBCA, and *ABCDCDAB*. But more complex knotted structures such as the pseudotrefoil are excluded from this class.

Akutsu & Uemura (A&U) Class

The set of pseudoknots that the *Akutsu*'s algorithm [Akutsu, 2000] can detect composes the *A&U class* of pseudoknots.

The basic components of the *A&U class* pseudoknots are the *simple pseudoknots* [Akutsu, 2000]. The terminology of Akutsu's simple pseudoknots contains two crossing stems, each with a set of base pairs. The right bases of the first stem and the left bases of the second stem are interleaved arbitrarily, and the other bases all lie outside the interleaved area, as shown in Figure 5.3(c). Particularly, the full circles in Figure 5.3(c) represent the right bases of the first stem, and the open circles represent the left bases of the second stem. Recursion allows the internal subfoldings of the unpaired strands, with or without pseudoknots, in the formed structure. In addition, Figure 5.3(d) shows an example of simple pseudoknot in the *A&U class*, which is representative as the corresponding shadow of the simple pseudoknot shown in Figure 2(A) in [Akutsu, 2000], with the pattern of *ABCBDADC*.

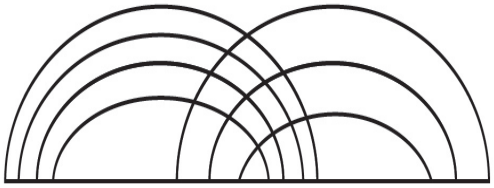
As a conclusion, the set of the *A&U class* of pseudoknots consists of any number of pseudoknot free structures, simple pseudoknots and their arbitrary concatenation and embedment inside each other [Nebel and Weinberg, 2012]. But more complex knotted structures with the interaction of three stems, such as kissing hairpin with the pattern of *ABACBC*, and complex pseudotrefoil with the pattern of *ABCABC*, are excluded from this class.

Jabbari & Condon (J&C) Class

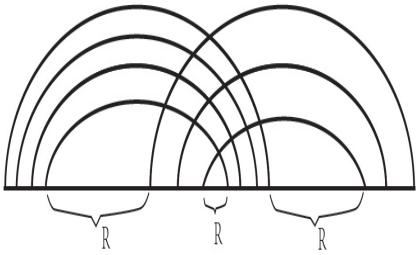
The set of pseudoknots that the *J&C*'s algorithm [Jabbari et al., 2007] can detect composes the *J&C class* of pseudoknots.

The basic component of the *J&C class* of pseudoknots is the set of the H-type pseudoknots and kissing hairpins, which are referred to as the *density-2 (D2)* pseudoknots.

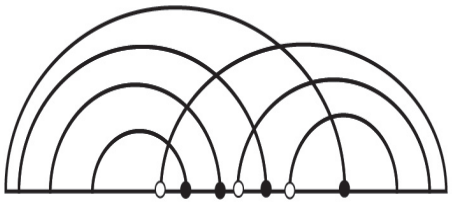
The notion of *density* is defined as the maximum number of stems where a par-



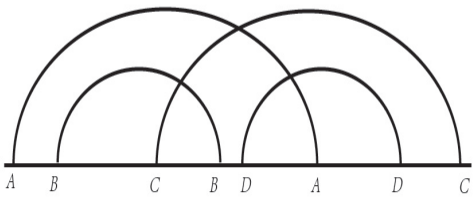
(a) An non-recursive H-type pseudoknot.



(b) A recursive H-type pseudoknot in the *D&P class*.



(c) The simple pseudoknot model of the *A&U class*.



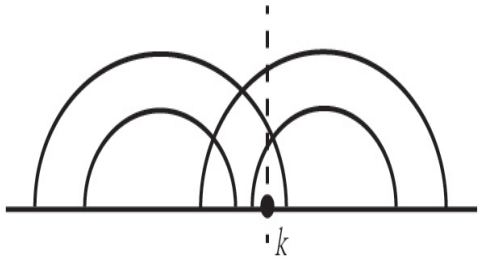
(d) A simple pseudoknot in the *A&U class*.

Figure 5.3: Algorithmic classification of pseudoknots.

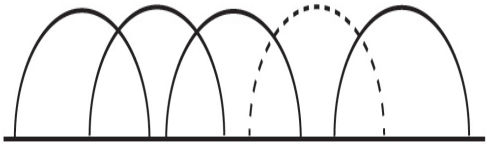
ticular nucleotide k is *embedded* or a vertical line through k is *intersected* [Jabbari et al., 2008], as shown in Figure 5.4(a). And the *J&C class* of D2 pseudoknots, corresponds to a class of pseudoknots in which each nucleotide is embedded in at most two stems, and recursions in length and depth are allowed, as shown in the Figure 5.4(b) and 5.4(c). Specially, the dashed line in both figures indicates an uncertain number of stems which are involved in an analogous fashion potentially.

Compared to the *A&U class* of pseudoknots, the kissing hairpins are included in the *J&C class* of pseudoknots, as the latter supports the conformation containing more than two stems. On the other hand, the simple pseudoknots in the *A&U class* with the pattern of *ABCBDADC*, as shown in Figure 5.3(d), are excluded from the D2 structures in the *J&C class*. The rejection of such pseudoknots is because

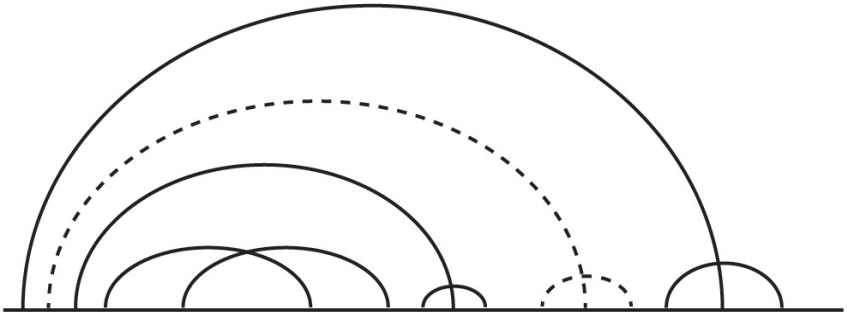
some nucleotides are embedded in more than two stems, such as the position k in Figure 5.4(a), where the vertical dashed line through k intersects three stems.



(a) An *A&U* simple pseudoknot corresponds to the density of 3, as the vertical line through k intersects three stems.



(b) A D2 structure with arbitrary number of stems in *J&C class*.



(c) A D2 structure with arbitrary depth of stems in *J&C class*.

Figure 5.4: The density and the *J&C class* of pseudoknots.

Last but not least, the *J&C class* contains the *L&P class* and the *D&P class* of pseudoknots, and is a subclass of the *R&E class* of pseudoknots, which is going to be introduced next.

Rivas & Eddy (R&E) Class

The set of pseudoknots that the *R&E*'s algorithm [Rivas and Eddy, 1999] can detect is the *R&E class* of pseudoknots.

The recursions of the gap matrices in the *R&E*'s algorithm allow the possibility of decomposing a large number of pseudoknots, making the *R&E class* of

pseudoknots the superclass of any other classes mentioned above [Condon et al., 2004]. This conclusion may be supported further by the results shown in Section 6.1.1.

Generally, the *R&E class* of pseudoknots consists of both *planar* pseudoknots and some *non-planar* pseudoknots, such as the pseudotrefoil with the pattern of *ABCABC*. An example which does not belong to the *R&E class* is the pseudoknot with the pattern of *ABCADCEDFEBF* [Rivas and Eddy, 1999].

The notion of planar pseudoknot defines the set of pseudoknots for which a planar representation does not require crossing lines. An example of planar pseudoknot, with the pattern of *ABCDCADB*, is shown in Figure 5.5, where the planar representation may not involve any crossing. In details, the nested base pairs *AA* and *CC* can be decomposed in the upper semi-plane, and the other two nested base pairs *BB* and *DD* can be decomposed in the lower one.

Particularly, the planar pseudoknots are also referred to as the *bi-secondary structures* in some other literature, with the definition of a superposition of two disjoint pseudoknot-free secondary structures [Haslinger and Stadler, 1999; Witwer et al., 2004].

Contrarily, the non-planar pseudoknots collect the pseudoknots for which a planar representation requires crossing lines. A typical example is the pseudotrefoil, with the pattern of *ABCABC*, as shown in Figure 5.6. Any of the base pairs *AA*, *BB* and *CC* crosses the other two. If the base pair *AA* is decomposed in the upper semi-plane, and *BB* is decomposed in the lower semi-plane, the base pair *CC* has to be decomposed in a third semi-plane to avoid the crossing with *AA* and *BB*, which is marked in the dashed line in Figure 5.6.

Quite obviously, all of the *L&P class*, the *D&P class*, the *A&U class*, and the *J&C class* mentioned above contain only planar pseudoknots.

Containments Between Classes

According to [Condon et al., 2004] and [Saule et al., 2011], the inclusion relation between these algorithmic classes, as well as the pseudoknot-free structures *(PKF)* and the arbitrary pseudoknots *(PK)* can be set as follows:

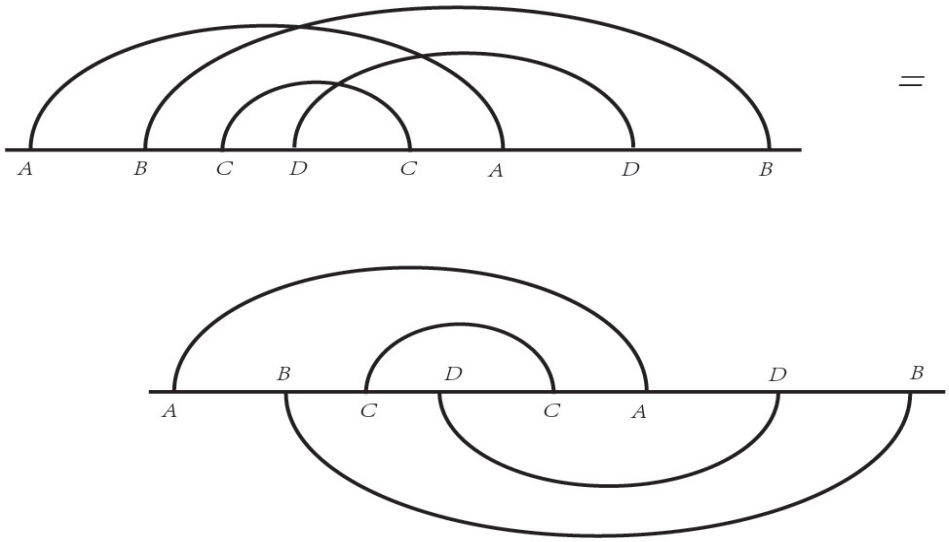


Figure 5.5: A planar pseudoknot with the pattern of *ABCDCADB*, which can be represented in a planar diagram.

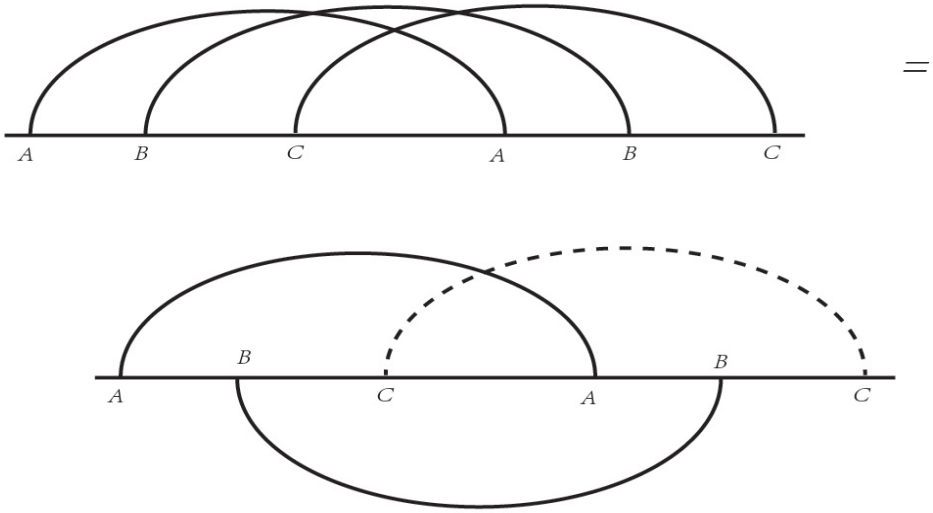


Figure 5.6: The non-planar pseudotrefoil with the pattern of *ABCABC*, which can not be represented in a planar diagram.

$$PKF \subset L\&P \subset D\&P \subset A\&U \subset R\&E \subset PK$$

where the *J&C class* of pseudoknots is ignored in this containment because of its incomplete inclusion with the *A&U class*. In other words, the *J&C class* and the

A&U class intersect partially.

Figure 5.7 and Table 5.1 conclude the description of the algorithmic classes of pseudoknots, where the *example* column in Table 5.1 shows a typical example that is not contained in all the classes before the current one. The *J&C class* of pseudoknots, taking a kissing hairpin as its corresponding example, is ignored in this conclusion for the same reason as above.

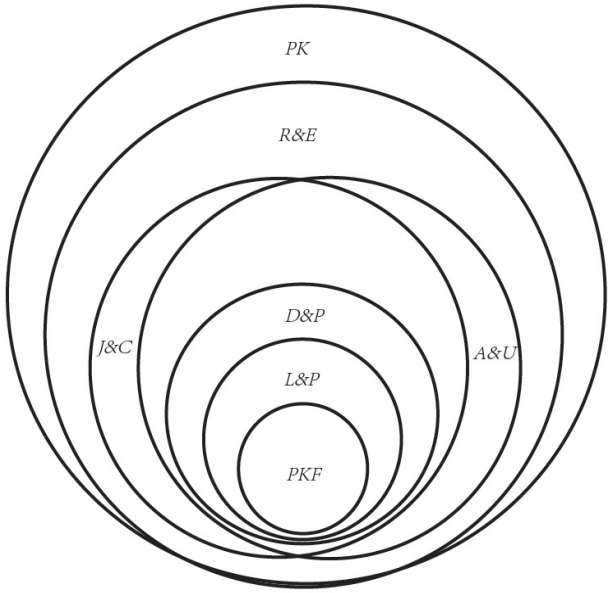


Figure 5.7: The Venn diagram of the algorithmic classes.

Table 5.1: The comparison of algorithmic pseudoknots

Class	Pseudoknot Models	Example
PKF	All nested structures	
L&P's	One H-type pseudoknot	ABAB
D&P's	Recursive H-type pseudoknots	ABCDCDAB
A&U's	Simple and recursive pseudoknots	ABCBDADC
R&E's	All planar and part of non-planar pseudoknots	ABCABC
PK	All planar and non-planar pseudoknots	ABCADBECDE

5.3.3 Conformational Characteristics

Besides the physical and algorithmic classifications, the pseudoknots can be classified according to some conformational or topological complexity measures,

such as the planar pseudoknots, or the bi-secondary structures, and the non-planar pseudoknots that are mentioned above. Further, this part is going to introduce three other conformational characteristics of pseudoknots, the *knot-component*, the *genus*, and the *page number*.

Knot-Component

[Rødland, 2006] defines the notion of *knot-component* as the structural components in the linear representation of RNA secondary structure, which are made by grouping the base pairs, with respect to some particular rules.

In detail, the knot-components collapse the consecutive base pairs in the pseudoknots, remove the nested substructures. Their illustration of the crossing interactions in the pseudoknots is quite analogous to that of the shadow of RNA secondary structures, as introduced in Section 5.3.1. But knot-components defined by [Rødland, 2006] correspond to structural elements, and the RNA shadow corresponds to a global secondary structure. We may say the knot-components are the bricks metaphorically, and the RNA shadow is a house built with these bricks.

Let P^n denote the knot-components with n stems, and $P^{n,k}$ distinguish the different types of knot-components with the same number of stems, where k is the numeration index proposed by the author. The classification is defined as:

- P^1: the pseudoknot-free structures consisting of only one stem, which is referred to as *orthodox* in this literature.
- P^2: the H-type pseudoknots, with the pattern of *ABAB*.
- $P^{3,1}$: the kissing hairpin pseudoknots, with the pattern of *ABACBC*.
- $P^{3,2}$: the pseudotrefoils, with the pattern of *ABCABC*.
- $P^{4,1}$: the complex pseudoknots such as the one with the pattern of *ABCBDADC*.
- $P^{5,1}$: the complex pseudoknots such as the one with the pattern of *ABCDEDBCAE*.

The knot-components may decompose the algorithmic classes of pseudoknots. Both the *L&P class* and the *D&P class* of pseudoknots only allow H-type pseudoknots, which belong to the P^2 type. The *A&U class* allows the H-type pseudoknots in the P^2 type, and complex pseudoknots in the $P^{4,1}$ type, but none of the others. The bi-secondary structures extend the *A&U class*, also allowing the kissing hairpin in the $P^{3,1}$ type additionally. The *R&E class* contains the pseudoknots in all of the P^2, $P^{3,1}$, $P^{3,2}$, $P^{4,1}$ and $P^{5,1}$ types [Rødland, 2006].

Genus

[Bon et al., 2008] recalls the notion of *genus* as the minimal number of handles that a surface should have, such that a diagram can be drawn on the surface without crossing. When the RNA secondary structure is represented by a double-line diagram in the linear model, as shown in Figure 5.8, the genus can be calculated by:

$$g = \frac{P - L}{2}$$

where P denotes the number of stems, and L denotes the number of closed loops, e.g. the number of closed circuit formed by the double lines.

Examples are given in Figure 5.8, where the base pairs in the left are replaced by double lines in the right, and the closed loops are highlighted in red. Specifically, the pseudoknot-free structure, as the first example, contains two stems and two closed loops. Both P and L equal to 2, suggesting the genus g-0. And both the latter two examples correspond to an H-type pseudoknot, with $P = 3$ and $L = 1$ respectively, suggesting the genus g-1.

There are four types of pseudoknots correspond to the genus g-1:

- The H-type pseudoknots, with the pattern of *ABAB*.
- The kissing hairpins, with the pattern of *ABACBC*
- The pseudotrefoil, with the pattern of *ABCABC*
- The complex pseudoknots, with the pattern of *ABCADBCD*.

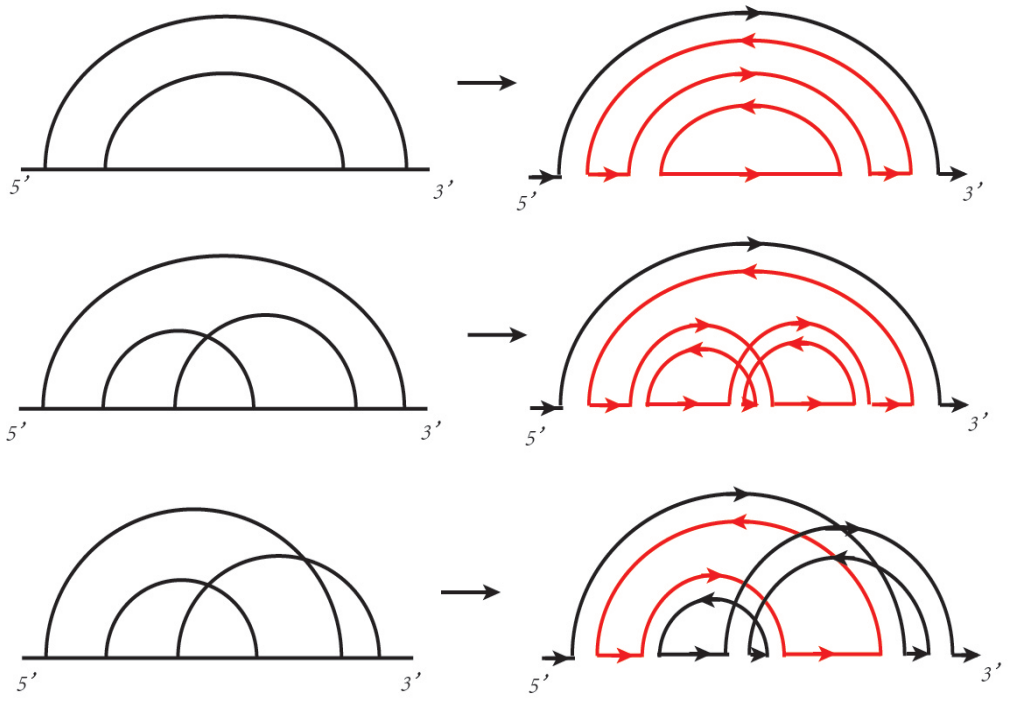


Figure 5.8: The schematic diagram of the double lines and closed loops (in red) in the calculation of genus, where the first one has a genus g-0, and the latter two have a genus g-1.

The pseudoknots having more complex interactions between the stems correspond to a genus g-2, such as the *delta ribozyme* RNA in the *hepatitis delta virus (HDV-It_g)*, with the pattern of *ABCDCADB* and the reference number of PKB75 in *PseudoBase*.

The pseudoknots classified by genus can be referred to as the γ-*structures* [Reidys et al., 2011]. The *0-structure* corresponds to a secondary structure without pseudoknots, and a *1-structure* corresponds to the four types of pseudoknots with the genus g-1. The *2-structures* and more general γ-*structures* have a similar correspondence.

[Reidys et al., 2011] compares the γ-structures with the algorithmic classes. The Venn diagram of the relations is shown in Figure 7 in [Reidys et al., 2011], with the conclusions:

- The *L&P class* and *D&P class* are subsets of the 1-structures.
- The *A&U class* and the 1-structures and 2-structures intersect partially.
- The *R&E class* may contain γ-structures with arbitrary γ. All the 1-

structures are contained in the *R&E class*, but a 2-structure with the pattern of *ABCADCEDBE* is an exclusion of the *R&E class*.

Page Number

A p-book is a set of p distinct half-planes, which are called as the *pages* of the book, that share a common boundary line l, which is called the *spine* of the book. The *book-thickness*, which is referred to as the *page number* of a graph, is the minimal number p of pages of a book into which it can be embedded so that the edges assigned to the same page do not cross. In the application of RNA secondary structures, the page number is the minimal number p such that the given secondary structure can be decomposed into a disjoint union of p nested substructures without crossing. [Clote et al., 2012; Haslinger and Stadler, 1999]

The notion of page number generalizes the structures which are either defined as the planar pseudoknots, with their page numbers of at most two, or classified into other classes. More precisely, the correspondence of the page number p to the pseudoknots is:

- $p = 1$: *1-page*, the pseudoknot-free structures.
- $p = 2$: *2-page*, the planar pseudoknots or the bi-secondary structures, such as the H-type pseudoknots with the pattern of *ABAB* and the kissing hairpins with the pattern of *ABACBC*.
- $p = 3$: *3-page*, complex pseudoknots, such as the pseudotrefoil with the pattern of *ABCABC*.
- $p \geq 4$: *4-page* and above, more general pseudoknots with more complex conformations. Some examples are shown in the results in Section 6.1.4.

The *L&P class*, the *D&P class*, the *A&U class* and the *J&C class* of pseudoknots are all *2-page* pseudoknots, while the *R&E class* pseudoknots may have page number $p \geq 3$.

Benchmark and Conformational Characteristics

As introduced in each conformational measurement above, the majority of natural pseudoknots have relatively low complexity values, such as the genus $g \leq 1$ or page number $p \leq 3$.

This benchmark is going to study the page number of each pseudoknot in the dataset manually. Although the calculation of page number for arbitrary pseudoknots has been proved to be NP-hard [Clote et al., 2012] and there is no precise solution in polynomial time, we can do this work thanks to the relatively low complexity of the natural pseudoknots which is proposed by [Clote et al., 2012].

We do not consider the other conformational characteristics in this benchmark for the following reasons. First, it is quite obvious that the bi-secondary structures correspond to the pseudoknots with page number of 2, thus the planar pseudoknots. Then [Bon et al., 2012] has already proposed an algorithm which both predicts a secondary structure and calculates a corresponding genus value for the given sequence. Last but not least, the Knot-Component classification separates the pseudoknots into classes according to the number of stems, which, for example, assigns an H-type pseudoknot and a kissing hairpin two different complexities. But both pseudoknots normally belong to the same complexity category under the other classifications, such as they are both planar pseudoknots, bi-secondary structures, pseudoknots with the genus g-1 and pseudoknots with page number of 2.

How Complicated Is the Calculation of Page Number?

The NP-completeness of computing the page number for arbitrary pseudokots is illustrated with the assistance of the *chromatic number* of a given secondary structure. The chromatic number of an RNA structure is defined by the minimal number of colors such that each base pair can be colored in a manner, with crossing base pairs in distinct colors [Clote et al., 2012].

In the context of the dot-bracket notation of RNA secondary structures as introduced in Section 2.3.1, the chromatic number, corresponding to the page number, can be represented by the number of different types of brackets used for

the given pseudoknot. In other words, the pseudoknot-free substructure in each page corresponds to the set of base pairs that are notated in the same type of brackets.

Saying given a shadow of an H-type pseudoknot, the page number is 2 as two types of brackets are enough to represent the crossing interactions of the given pseudoknot, the parentheses and the square brackets. But given such complex pseudoknots with the pattern of *ABACDCEBED*, how about their page number? And how about the page number for more general ones?

In this part of work, this benchmark considers to utilize as less types of brackets in as many cases as possible. In other words, this idea can be realized by checking the availability of all the brackets in hand iteratively for each base pair in the given pseudoknot. And a new type of brackets is introduced until all the previous types are not available to notate the crossing interactions any more. More precisely, it is supported by prioritizing the types of brackets, and assigning the foremost type of brackets available to the current base pair.

Table 5.2 shows some examples about this idea of *saving the types of brackets for page number*, where the parentheses '(' and ')' and square brackets '[' and ']' are used to represent a pseudoknot initially, and the parentheses hold the highest priority of notation. The curly brackets '{' and '}' are introduced when the former two types of brackets are incapable to represent the conformation, and later the alphabetical letters 'A', 'a' and 'B', 'b' are introduced similarly.

In fact, the operation upon this idea of *saving the types of brackets for page number* works well for most cases, out of the 414 pseudoknots in this benchmark. It follows the order of picking the parentheses for the current base pair whenever they are available, and then the square brackets, and then the curly brackets, and so forth.

However, for the complex pseudoknot *c2* in Table 5.2, is its page number equal to 3? No, the answer is 2, but with the corresponding dot-bracket notation breaking the predefined prioritization. The final notation of *c2* which is highlighted in red shows that *c2* also has the possibility of being represented in two types of brackets, corresponding to a page number of 2.

In fact, the problem of assigning a proper dot-bracket notation to the given

Table 5.2: The page number of some typical pseudoknots.

Example	Page Number	Dot-Bracket Notation		Prioritization
H-type	Page No.	Pattern	ABAB	
	2	Notation	([)]	
Kissing hairpin	Page No.	Pattern	ABACBC	
	2	Notation	([)(])	
A recursive	Page No.	Pattern	ABCDCDAB	'('and ')'
	2	Notation	([([)])]	First
Pseudotrefoil	Page No.	Pattern	ABCABC	
	3	Notation	([{)]}	
A complex c1	Page No.	Pattern	ABCDCADB	
	2	Notation	([([)]]	
A complex c2	Page No.	Pattern	ABCDBDECAE	
	3 ?	Notation	(([[]]{])}	'(',')' First
	2	Notation	[([[)](]])	A notation.

pseudoknot with a minimal page number is not the question of predefining the order of choosing the brackets, but a foresight to the crossing interactions globally. The base pair *EE* in *c2* crosses the base pairs *AA* and *CC*, and *CC* crosses the base pair *BB*. This makes the base pair *BB* have to hold a different bracket type from the base pair *AA* for the sake of saving the types of brackets, although *AA* and *BB* are nested.

This illustrates, in some sense, the NP-hardness of computing the page number. If $P \neq NP$, there is no polynomial algorithm can compute it in the general cases.

However as the page number of the pseudoknots in nature is relatively low, this benchmark concludes all the page number for the 414 pseudoknots in the Section 6.1.

5.4 Methods Involved

This benchmark is going to consider 11 programs with different options on the algorithms. If the program allows an alternative algorithm, it is referred to as a different method.

Totally, there are 15 methods: *CyloFold*, *DotKnot*, *DotKnot-K* representing DotKnot with the kissing hairpin algorithm, *HotKnots-cc* representing HotKnots

with the CC energy model, *HotKnots-dp* representing HotKnots with the DP energy model, *HotKnots-re* representing HotKnots with the RE energy model, *IPknot*, *MC-Fold*, *McGenus*, *McQFold*, *pKiss*, *pknotsRG-M* representing pknotsRG with the MFE algorithm, *pknotsRG-F* representing pknotsRG with the enforcing pseudoknots algorithm, *pknots*, and *vsfold5*.

Typically, the variants of IPknot used in Chapter 4, as shown in Section 4.4.3, are slightly different with the one used in this benchmark, which is caused by the fashion of the utilization. In detail, IPknot-2 and IPknot-3 denote two decomposed levels employed by the web service of IPknot [Sato et al., 2011] in Section 4.4.3, and IPknot in this benchmark corresponds to the single algorithm of the locally installed version.

The listed methods in this benchmark are not exhaustive. Some other methods are not taken into account as the unavailability of their executables or editable outputs, or in consideration of some other reasons. These programs and more explanations are shown in Section 5.4.3.

5.4.1 Exact Methods

pknots

pknots implements the *R&E*'s algorithm [Rivas and Eddy, 1999], which is elaborated in Section 3.2.1 as a typical exact method predicting pseudoknots, with the assistance of the dynamic programming.

As a pioneer of the prediction of RNA secondary structure including pseudoknots, pknots opened the door to the world of predicting pseudoknots based on the idea of maximizing the thermodynamic stability of the conformation. The algorithm has the complexity of $O(n^6)$ in time and $O(n^4)$ in space, and captures a fairly general class of pseudoknots.

pknotsRG

The *R&G*'s algorithm [Reeder and Giegerich, 2004], *pknotsRG*, is designed based on the MFE model and the dynamic programming. The complexity is $O(n^4)$ in time and $O(n^2)$ in space.

The algorithm calculates the MFE structures based on the model of the *canonized simple recursive* pseudoknots principally, which allow the crossing interaction of two stems and the arbitrary internal interaction of unpaired strands surrounded by the pseudoknots.

The canonization policy is employed to restrict the search space of MFE structures. And there are three restrictions on the canonization of the *simple recursive* pseudoknots, as proposed in the *A&U class* of pseudoknots: both stems must not have bulges, both stems must have maximal extent, and the compartment *Loop 2* must not be negative as both stems compete for the same bases of *Loop 2* for a maximal extent.

The *R&G*'s algorithm provides three variants of predicting pseudoknots:

- *pknotsRG-mfe*, which computes the MFE structure, with or without pseudoknot.
- *pknotsRG-enf*, which picks out the energetically best structure with pseudoknot from the folding space.
- *pknotsRG-loc*, which computes the energetically best pseudoknot formed in a local region in the sequence, the one has the best energy to length ratio.

The variants of *pknotsRG-mfe* and *pknotsRG-enf* are both considered in this benchmark, which are referred to as two different methods of the *pknotsRG-M* and the *pknotsRG-F* respectively. But the *pknotsRG-loc* is ignored since the unavailability of its global conformation.

5.4.2 Heuristic Methods

HotKnots

[Ren et al., 2005] reports an heuristic algorithm, *HotKnots*, for predicting pseudoknots. Roughly, the algorithm builds up candidate secondary structures by adding low-energy substructures one at a time to partially formed structures based on the thermodynamic model extended for pseudoknots as in [Dirks and Pierce, 2003].

The calculation is remained as a tree since multiple partially formed structures are maintained and each of them considers several different additions of a single substructure. The added substructures are termed as *hotspots* which are energetically favorable structural elements determined by [Zuker and Stiegler, 1981] with the constraint that no base already paired may be present in the structure.

In detail, a set of hotspots is built up in a tree like fashion, and each hotspot in the set is used as a basis for expanding a secondary structure for the given sequence. The output of the algorithm is a list of secondary structures corresponding to each hotspot set, sorted by their free energies.

Besides the thermodynamic model extended from the *D&P*'s algorithm, HotKnots still uses the energy parameters from two energy models for secondary structures with pseudoknots, the *R&E*'s model [Rivas and Eddy, 1999], and the *Cao&Chen (CC)*'s model [Cao and Chen, 2006]. Specifically, the computation of HotKnots with the three energy models are referred to as three methods in this benchmark, notated as *HotKnots-dp*, *HotKnots-re* and *HotKnots-cc* respectively.

vsfold5

vsfold5 is an algorithm predicting MFE pseudoknots by using structure mapping and an entropy model, along a sequential, from 5' end to 3' end, and thermodynamically plausible folding pathway [Dawson et al., 2007].

The folding pathway is described as the decomposition of the structure into a set of substructures. And the pseudoknots are considered as the addition of stems into the loop region of a formed stable secondary structure. And typically, vsfold5 employs the mapping routines of the *pointers* for the secondary structure without pseudoknots, and *handles* for the pseudoknots.

In details, in the globular model of vsfold5, a pointer consists of the current base pair, a tag suggesting the structural element where the base pair locates, a forward link which is used to map the next base pair of the secondary structure from the current one, and a reverse link.which is used to map any previous part that is present in the structure. While, a handle, as the extension of the pointers, contains the indexes considering more complex and detailed information additionally to map out the global configuration.

MC-Fold

MC-Fold is a phase of the pipeline proposed in [Parisien and Major, 2008] which infers the secondary and tertiary structures from the given sequence.

Based on the *nucleotide cyclic motifs (NCM)*, MC-Fold predicts a sorted list of possible secondary structures for the given sequence according to their thermodynamic stabilities. The predicted secondary structures can be sent into the next phase *MC-Sym* of the pipeline that determines the RNA three-dimensional structure.

More precisely, the NCMs are two types of cyclic structural elements, the lone-pair loops, corresponding to the hairpin loops, that are up to six nucleotides, and double-stranded NCMs, corresponding to stems, bulges and interior loops, that are up to eight nucleotides.

MC-Fold determines a list of initiation sites which are assigned lone-pair NCMs, and then recursively matches the rest of the given sequence to the double-stranded NCMs. Finally, MC-Fold determines a set of assemblies of the stem-loops for the given sequence, and ranks them according to their free energies.

McQFold

McQFold is a probabilistic model for predicting RNA secondary structures with pseudoknots, it employs a *Markov-chain Monte-Carlo (MCMC)* method for sampling RNA structures in the Bayesian framework, according to their posterior probability distribution for a given sequence [Metzler and Nebel, 2008].

The basic idea of the algorithm is to use a *stochastic context-free grammar (SCFG)* to generate a pseudoknot-free framework of a structure. And then, the algorithm additionally develops a special symbol q, as a terminal in the grammar, in order to generate pairs of regions in the sequence that will form the pseudoknots further.

The main idea of McQFold is analogous to HotKnots, which appends the additional stems to the partially formed structure, but according to the probability of base pairs, rather than the thermodynamic stability in HotKnots.

CyloFold

CyloFold simulates a folding process by choosing stems based on the established energy rules and using a three-dimensional model for representing the RNA structures [Bindewald et al., 2010].

The idea of CyloFold is to maximize matching helices in a secondary structure. Initially, CyloFold generates a stem list of all possible stems with more than three base pairs. And the secondary structure prediction is performed by picking the structure from the stem list with best score, where the score is set to be the sum of the free energy of the already placed stems. And fifty rounds of the folding simulations are performed to return the overall optimal structure.

DotKnot

DotKnot predicts the RNA pseudoknots by extracting stem regions from the secondary structure probability dot plot and assembling pseudoknot candidates [Sperschneider and Datta, 2010].

The basic idea of DotKnot is to calculate the secondary structure partition function first, as in [McCaskill, 1990], in the purpose of finding a set of promising structure elements in $O(n^3)$ of time and $O(n^2)$ of space, which may contain potential pseudoknot foldings. Second, DotKnot assembles the pseudoknot candidates using this set of promising elements in two levels, finding the stable core H-type pseudoknots, and then the recursive formations. Last, DotKnot employs the loop entropy parameters to evaluate their free energy values and credibility in the folded sequence. The output of DotKnot is a set of detected pseudoknots and a global conformation.

Additionally, DotKnot considers the prediction of pseudoknots including the kissing hairpins, which is referred to as the variant method *DotKnot-K* in this benchmark.

pKiss

As the successor of pknotsRG, [Theis et al., 2010] proposes an heuristic method for predicting RNA pseudoknots including kissing hairpins, *pKiss*. pknotsRG

considers the class of canonized simple recursive pseudoknot, while pKiss considers the canonized simple recursive kissing hairpins specifically.

In details, the basic idea of pKiss is the view that the kissing hairpins can be referred to as an overlap of two simple pseudoknots, as shown in Figure 5.9. Consequently, pKiss uses a way similar to pknotsRG to predict an optimal simple pseudoknot s_1 as the left pseudoknot in the overlap. And then it searches for another simple pseudoknot s_2, such that the left part of $s2$ may match the right part of the previously computed s_1, e.g. the BBs, and with the 5' end of the CC of s_2 lying strictly to the right of the 3' end of the AA of s_1. A symmetric step starting from predicting an optimal choice as the right pseudoknot in the overlap is applied in a second round. The output of pKiss is the energetically better prediction between the two rounds of detections.

pKiss also supports other strategies of predicting kissing hairpins, but this benchmark considers the detection introduced above, which is referred to as Strategy A, pKiss's default mode of predicting pseudoknots.

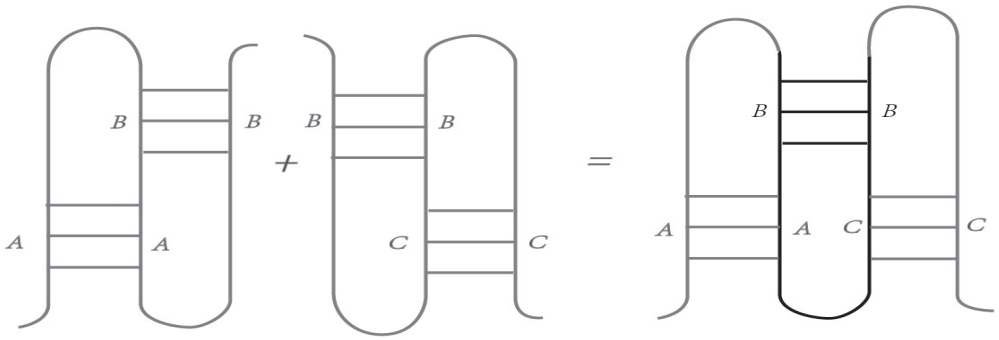


Figure 5.9: The overlap of two H-type pseudoknots in the pKiss's model.

IPknot

IPknot predicts RNA secondary structures with pseudoknots based on maximizing expected accuracy of a predicted structure with respect to an ensemble of all possible structures [Sato et al., 2011].

Similar to most approaches predicting pseudoknots, IPknot decomposes a pseudoknotted structure into a set of pseudoknot-free substructures and approximates

the base-pairing probability distribution that considers pseudoknots, which is used in the following integer programming objective function. And the maximization of expected accuracy refers to maximizing the expectation of the number of true predictions of base pairs under the computed probability distribution.

The solution of the integer programming problem corresponds to the thermodynamically optimal pseudoknotted prediction of IPknot.

McGenus

McGenus [Bon et al., 2012] uses a Monte Carlo algorithm to search for an MFE structure, with a general scoring function which includes both the free energy contributions for pair stacking, loop penalties, etc. and a penalty for the topological genus of the pseudoknots.

An RNA structure in the McGenus's model is referred to as a collection of stem-like structures, which are termed as the *helipoints*, the ensemble of helices, or stems, for a given sequence. The MFE structure amounts to the set of pairwise compatible helipoints for which the overall free energy is minimum. The compatibility of two helipoints arise when there is no shared base between them. And the choosing of the helipoints is done according to the stochastic Monte Carlo scheme. The output of McGenus is a set of pseudoknots marked with the corresponding genus.

5.4.3 Benchmark and Prediction Methods

Table 5.3 presents all the 15 methods considered in this benchmark in alphabetic order. There still are other approaches predicting RNA pseudoknots from a single sequence.

KnotSeeker [Sperschneider and Datta, 2008] uses a hybrid sequence matching and free energy minimization approach to perform a screening of the sequence. The short sequence fragments are considered as possible candidates that may contain pseudoknots, suggesting the output of KnotSeeker is a set of partial pseudoknots found on the sequence fragments rather than a global structure based on the entire sequence. Lots of the omitted structural information between the pseudoknots make KnotSeeker excluded from the prediction methods of this benchmark,

although it is available to handle long sequences.

HFold [Jabbari et al., 2007], the corresponding program of the *J&C*'s algorithm, is not included in this benchmark because its prediction is restricted by the pseudoknot-free structure provided as input, as introduced in Section 3.2.1. In this benchmark, we only consider the prediction methods which take a single sequence as input, and yield a secondary structure or several ones as output.

In addition, some programs are omitted in this benchmark as the unavailability of the executables, such as *FlexStem* [Chen et al., 2008], *HPknotter* [Huang et al., 2005] and *TT2NE* [Bon and Orland, 2011][1]. Some are ignored as either there are no editable outputs, such as *Kinefold* [Xayaphoummine et al., 2005] and *ProbKnot* [Bellaousov and Mathews, 2010], or the incapability of compiling the executables, such as the *iterative loop matching approach (ILM)* [Ruan et al., 2004].

5.4.4 Normalization of the Predictions

As mentioned in Section 2.3.1, the methods may have different formats of output of RNA secondary structures, either be in dot-bracket notations, or in BPSEQ format, or in CT format. As a result, this part is going to introduce the idea of the normalization of the predictions in the purpose of comparisons.

Specifically, there are two types of normalization, translating the dot-bracket notations into the BPSEQ or CT formats, and the reverse operation. The former translation is needed for calculating the evaluation values which are introduced in the next section. And the latter is needed for facilitating the intuitive comparison of the predictions.

In fact, the evaluation of the predictions, compared to the reference structures, is carried out by comparing both structures in the BPSEQ format. It is relatively easy to translate the predictions from the CT format into the BPSEQ format, by just removing the third, fourth and sixth columns from the original files. On the other hand, translating the prediction from the dot-bracket notation into the

[1]It is an algorithm of predicting MFE pseudoknots based on the topological genus classification. But as TT2NE is available only with the web service and functionally analogous to McGenus, it is not considered in this benchmark.

Table 5.3: The 15 methods considered in the benchmark.

	Methods	Utilization	Version	Date	Reference
Heuristic Methods	CyloFold	Web Service			[Bindewald et al., 2010]
	DotKnot	Local Installation	1.3.1	Oct. 2011	[Sperschneider and Datta, 2010]
	DotKnot-K	Local Installation	1.3.1	Oct. 2011	[Sperschneider and Datta, 2010]
	HotKnots-cc	Local Installation	2.0	Jan. 2010	[Ren et al., 2005]
	HotKnots-dp	Local Installation	2.0	Jan. 2010	[Ren et al., 2005]
	HotKnots-re	Local Installation	2.0	Jan. 2010	[Ren et al., 2005]
	IPknot	Local Installation	0.0.2	Jan. 2011	[Sato et al., 2011]
	MC-Fold	Remote Server[1]			[Parisien and Major, 2008]
	McGenus	Local Installation		Feb. 2013	[Bon et al., 2012]
	McQFold	Local Installation		May 2006	[Metzler and Nebel, 2008]
	pKiss	Local Installation	2.2.11	Dec. 2014	[Janssen and Giegerich, 2014]
	vsfold5	Web Service	5.23		[Dawson et al., 2007]
Exact Methods	pknotsRG-M	Local Installation	1.3	Sep. 2006	[Reeder and Giegerich, 2004]
	pknotsRG-F	Local Installation	1.3	Sep. 2006	[Reeder and Giegerich, 2004]
	pknots	Local Installation	1.08	Sep. 2012	[Rivas and Eddy, 1999]

[1] MC-Fold is provided with the .cgi file, which computes the structure for a given sequence on the remote server. And the version information is unknown.

BPSEQ format can be done with an algorithm which uses several stacks.

More precisely, the number of different types of brackets represented in the dot-bracket representation corresponds to the number of stacks used. And each stack is used to store one distinct type of opening brackets.

Given a secondary structure S with n nucleotides in the dot-bracket representation, the translation starts from the 5' end of the sequence to the 3' end. A dot '.' in the S is referred to as a '0' in the third column of this corresponding unpaired base in the generated BPSEQ file. The position i of a opening parenthesis '(' in S is deposited into the first stack, the position of a opening square bracket '[' is deposited into the second stack, the position of a opening curly bracket '{' is deposited into the third stack, and so forth.

The encounter of a closing parenthesis ')' will pop the uppermost element of the first stack, and both positions of the closing parenthesis and its popped partner will be stored into the generated BPSEQ file as a base pair. Similarly, the encounter of a closing square bracket ']' or a curly bracket '}' will have an analogous operation on popping the top element of the second or third stack, and having both positions of the matching pair of brackets deposited into the BPSEQ file as a base pair. And so forth.

However, we hardly capture the crossing interactions from the BPSEQ files readily, and decide to devote more efforts into the opposite operation. We believe the translation of structures from the BPSEQ file into the dot-bracket notation can intuitively facilitate the comparison of the predictions, as the results shown in Table B.2.

But, how to describe the pseudoknotted conformation properly in the dot-bracket representation? The utilization of a set of stacks may answer the question, where each stack is used to store the base pairs that can be represented in one particular type of brackets.

Given a BPSEQ file S, an initialization step is to remove the unpaired bases from S, and generate the BP, a list of base pairs (x, y), where x and y are the first and third columns of each base pair in S. The second column of the sequence information is omitted temporarily, as it is easy to be referred later with the corresponding position of x. The procedure of processing the BP is to consider

each base pair in the ascending order of x. The global idea is to either deposit the current base pair in one of the stacks, or pop the top element of a stack and store the base pair in the result list Str, which represents the secondary structure in the dot-bracket notation.

As the preliminaries, here are two important notions in the process:

- The *crossing* of two base pairs. Given two base pairs (x_1, y_1) and (x_2, y_2) with $x_1 < y_1$ and $x_2 < y_2$, they are crossing if the conditions of either $x_1 < x_2$, $y_1 < y_2$ and $x_2 < y_1$, or $x_2 < x_1$, $y_2 < y_1$ and $x_1 < y_2$ are satisfied.

- The *matching* of two base pairs. Given a base pair (x, y), the matching of two base pairs is declared if the pair (y, x) is encountered.

Initially, the first base pair is deposited in the first stack, which is particularly used to store the base pairs that can be represented in parentheses '(' and ')' in Str. And then the second base pair (x, y) will be checked with the possibilities that whether (x, y) matches the top element of the first stack, i.e. the first base pair, or crosses it. If they are matched, saying the top element of the first stack is the base pair (y, x), the (y, x) will be popped, and stored in the Str with assigning y a '(' and x a ')'. If they are crossed, a new stack is desired. Consequently, (x, y) is going to be deposited in the second stack, declaring that (x, y) and its future 'stack-mates' are going to be represented in square brackets '[' and ']' in Str. If both possibilities fails, (x, y) is deposited in the current stack, as it is compatible with the element of the first stack.

The third base pair (x, y) will be checked with the possibilities of matching and crossing with the top elements of the stacks, if there are more than one stack. If there is a matching, the matched (y, x) will be popped, and y will be assigned a '(' or '[' depending on which stack the (y, x) is found, and x will be assigned a ')' or ']' accordantly. If (x, y) crosses the top elements of both stacks, it may be deposited in the third stack, with its representation and future stack-mates' in Str of the '{' and '}'. If neither is satisfied, (x, y) can be deposited either in the first stack if it is compatible with the element of the first stack, or in the second stack if not. And so forth.

Algorithm 2 concludes this idea of translating the secondary structure from the BPSEQ file into the dot-bracket representation.

We may notice that the first stack, which holds the parentheses for the representation of its members in Str, has the highest priority of storing the base pair (x, y), if (x, y) is compatible with more than one stack. The next priority goes to the second stack, and so forth. This caters to the idea of *saving the types of brackets for page number* which is mentioned in the Section 5.3.3, with the same prioritization of choosing the brackets for base pairs.

We expect such kind of prioritization may reduce the number of stacks used, which reflects the number of different types of brackets used, and further the page number of the given structure. But the efforts may not succeed with the inaccessibility of any foresight to the global crossing interactions and other inestimable complex reasons, as mentioned above, although the expectation is supported by the study of page number for the 414 pseudoknots in this benchmark.

5.5 Evaluation Parameters

In the purpose of evaluating the performance of the predictions by 15 methods, this benchmark is going to use the same criteria as in Chapter 4: the *sensitivity*, the *positive predictive value (PPV)* and the *Matthews Correlation Coefficient (MCC)*.

The computation of the three evaluation values are given again, with the TP, FP, TN and TP having the same definitions as in Section 4.3.1.

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.1}$$

$$PPV = \frac{TP}{TP + FP} \tag{5.2}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5.3}$$

And as the participation of crossing interactions, the Equations 5.2 and 5.3 do not consider the division of false positive further, for the reason explained in Section 4.3.2.

Algorithm 2 The algorithm of translating the structures from BPSEQ to dot-bracket representation.

Input: A BPSEQ file S for a given structure, either containing the pseudoknot(s) or not.

Output: A list Str, storing the structure in the dot-bracket representation.

1: **procedure** TRANSLATION(BP)

Initialization: Remove the unpaired bases from S, and assign a dot to the corresponding position in Str. Create BP, the list of base pairs (x, y) of S, in the ascending order of x. Create bpS, a list of stacks. Create a stack, and add it to bpS.

2: $n \leftarrow$ size of bpS.
3: **for** each (x, y) in BP **do**
Loop:
4: **for** $i \leftarrow 1$ to n **do** ▷ iteratively check all the stacks to deposit (x, y).
5: **if** $bpS(i)$ is empty **then**
6: **if** $bpS(i)$ is the only stack **then**
7: Push (x, y) into $bpS(i)$.
8: Invoke *STACK-SYM* to assign a symbol to $bpS(i)$.
9: Break Loop.
10: **else** ▷ $bpS(i)$ is not the only stack in bpS.
11: **for** $j \leftarrow i + 1$ to n **do**
12: /*Check if (x, y) matches with the top element of the
13: following stack(s) if the current stack is empty.
14: Either pop the match if yes, or deposit it into the empty stack if no.*/
15: **if** $bpS(j)$ is not empty and (x, y) matches the top element of $bpS(j)$ **then**
16: Pop the top element of $bpS(j)$.
17: Assign the symbol of $bpS(j)$ to y, invoke *ASSIGN-BRACKETS* to
18: assign a closing bracket to x, and add them to Str.
19: Break Loop.
20: **end if**
21: **end for**
22: Push (x, y) into $bpS(i)$. ▷ Push (x, y) into current stack.
23: Break Loop.
24: **end if**

```
25:         else                                          ▷ bpS(i) is not empty.
26:             if (x, y) matches the top element of bpS(i) then
27:                 Pop the top element of bpS(j).
28:                 Assign the symbol of bpS(i) to y, invoke ASSIGN-BRACKETS to
29:                 assign a closing bracket to x, and add them to Str.
30:                 Break Loop.
31:             else if (x, y) crosses the top element of bpS(i) then
32:                 if i = n then                         ▷ New stack is desired.
33:                     Create a new stack, and add it to bpS.
34:                     Push (x, y) into the new stack.
35:                     Invoke STACK-SYM to assign a symbol to bpS(i + 1).
36:                     Break Loop.
37:                 else
38:                     Continue Loop.                    ▷ Check the availability of the
39:                                                       ▷ next stack to deposit (x, y).
40:                 end if
41:             else
42:                 /*In the case that (x, y) does neither match nor cross
43:                 the top element of bpS(i), push (x, y) into it.*/
44:                 Push (x, y) into bpS(i).
45:                 Break Loop.
46:             end if
47:         end if
48:     end for
49: end for
50: end procedure
```

```
1: procedure STACK-SYM:(Number)
2:     return a opening parenthesis to the stack in response to its position Number in bpS.
3:     Specifically, '1' corresponds to the first level of brackets, a '(', '2' to a '[', '3' to a '{'.
4:     And then '4' and '5' corresponds to the alphabetical letters 'A' and 'B', representing
5:     a higher level of crossing, and so forth.
6: end procedure
```

```
1: procedure ASSIGN-BRACKETS(Character)
2:     return a closing parenthesis, in response to the opening parenthesis Character.
3: end procedure
```

Chapter 6

Results

6.1 Pseudoknot Classification

6.1.1 Global Classification

According to the three classifications mentioned in Section 5.3, the classification of the 414 pseudoknots is shown in Table 6.1. Particularly, the third category of *Algorithmic Accessibilities* shows the algorithmic classifications which have been computed by the software *RNAtest*, provided by Condon et al. [Condon et al., 2004] kindly. A Y is assigned when the current pseudoknot falls into the certain algorithmic class, and an N represents the opposite.

Additionally, Section 5.3.2 shows an inclusion relation between the classes, except the *J&C class*. Therefore, the number of pseudoknots in each algorithmic class, and the number of pseudoknots which are in the complementary set of the current class compared to its superset are shown respectively in Table 6.1. The complete information of the classification for the 414 pseudoknots is provided in Appendix C, and the corresponding details of each sequence, such as their RNA type, organism, sequence and structure, are shown in Supplementary File *Benchmark*.

6.1.2 Correlation between the Classifications of Sequences

The last section has shown the classifications of 414 RNA pseudoknots. Meanwhile, as mentioned in Section 5.1, we are considering a particular collection of

Table 6.1: The classification of the 414 pseudoknots.

Physical Interactions					
H-type		Kissing	Recursive	Complex	
341		25	4	44	
Conformational Characteristics					
Page No.=2		Page No.=3		Page No.=4	
409		3		2	
Algorithmic Accessibilities					
L&P class	*D&P class*	*A&U class*	*J&C class*	*R&E class*	Number
The Number of Pseudoknots Belonging to Each Algorithmic Class					
Y					333
	Y				344
		Y			344
			Y		370
				Y	411
The Number of Pseudoknots in the Complementary Set of Each Class					
Y	Y	Y	Y	Y	333
N	Y	Y	Y	Y	11
N	N	Y	Y	Y	0
N	N	N	Y	Y	26
N	N	N	N	Y	41

pseudoknots which contains the sequences that can be handled by all the methods, with the consideration of comparing the predictions of all the methods with a consistent set of sequences. And 387 sequences compose the *shared set*, with the detailed reasons of selection given in Section 6.2.

In order to have a comprehensive understanding of the relation between the characteristic of sequences and the classification of pseudoknots of the 387 sequences, we are going to introduce the *correlation*. Typically, the notion of the correlation is to find the relationship between different classes of sequences, which are divided into subsets in accordance to diverse aspects of the sequences, such as the length, organism and page number of them. This study will benefit the comparison of the prediction methods based on the shared set, as the methods are going to be compared based on these individual classes of sequences.

In practice, we are going to count how many sequences in a particular class intersect with another one. And as the functional families of the RNA sequences are the most interesting parts for the bioinformatics community, we show the result in Table 6.2, where the numeric values represent the number of sequences belonging to the corresponding two classes.

6.1.3 The Recursive and Complex Pseudoknots

The H-type pseudoknots and kissing hairpins have well-known crossing interactions of *ABAB* and *ABACBC* respectively. And this dissertation pays more attention on the four recursive pseudoknots and 44 complex pseudoknots in Table 6.1, which are provided in detail in Tables 6.3 and 6.4.

Specially, the pseudoknot pattern of some complex pseudoknots with page number ≥ 3, as shown in Table 6.4, is assigned *others* as their much more complicated crossing interactions. Typically, a majority of the complex pseudoknots in PseudoBase are the ribozymes from the eukaryotic molecules. The homology of these sequences contribute them a same pseudoknot pattern, a same page number and the same affiliation with the algorithmic classes in Table 6.4. The akin homology of 3RKF_A, 3IVN_B and 3LA5_A is shown as well. And typically, as the inaccessibility of the functional family of 3KIY_A, we may not conclude the homology between 3KIY_A and 2WDL_A.

Table 6.2: The correlation between the classifications of sequences.

		Aptamers	mRNA	tRNA	tmRNA	rRNA	Riboswitch	Ribozymes	Others
Length	≤ 100 nt	14	6	8	10	5	10	31	11
	101-160 nt	1	10	0	0	5	1	6	2
Organism	Eukaryote	1	8	2	0	7	0	29	10
	Prokaryote	5	5	5	10	3	7	3	2
	Virus	0	2	0	0	0	0	4	0
	Unknown	9	1	1	0	0	4	1	1
Page No.	2	15	15	8	10	10	11	37	13
	3	0	1	0	0	0	0	0	0
Pknot Type	H-type	13	15	0	10	6	5	5	12
	Kissing	1	0	8	0	4	3	2	1
	Complex	1	1	0	0	0	3	30	0

		Vr. 3 UTR	Vr. 5 UTR	Frame shift	Vr.Read through	Vr. tRNA-like	Vr. Others	Unknown
Length	≤ 100 nt	103	29	30	7	52	23	6
	101-160 nt	0	0	3	0	6	7	1
Organism	Eukaryote	0	0	2	0	0	0	0
	Prokaryote	0	0	0	0	0	0	0
	Virus	103	29	31	7	58	30	2
	Unknown	0	0	0	0	0	0	5
Page No.	2	103	29	33	7	58	30	7
	3	0	0	0	0	0	0	0
Pknot Type	H-type	102	29	33	7	58	30	5
	Kissing	1	0	0	0	0	0	2
	Complex	0	0	0	0	0	0	0

Table 6.3: The 4 recursive pseudoknots.

Name	Length	RNA Type	Organism	Pseudoknot Type	Pseudoknot Pattern	Page No.	*L&P class*	*D&P class*	*A&U class*	*J&C class*	*R&E class*
3IYQ_A	349	tmRNA	Thermus thermophilus HB8	recursive H-type	ABAB*	2	N	Y	Y	Y	Y
3IZ4_A	377	tmRNA	Escherichia coli	recursive H-type	ABAB*	2	N	Y	Y	Y	Y
3J2C_N	927	16S rRNA	Escherichia coli	recursive H-type	ABAababcdcdB	2	N	Y	Y	Y	Y
3JYX_5	3170	26S rRNA	Thermomyces lanuginosus	recursive kissing hairpin	ababABAcdcdDEDFEFefefCBC	2	N	N	N	Y	Y

* But this pseudoknot contains four identical pseudoknots, all with the pattern of *ABAB*.

Table 6.4: The 44 complex pseudoknots.

Name	Length	RNA Type	Organism	Pseudoknot Type	Pseudoknot pattern	Page No.	*L&P class*	*D&P class*	*A&U class*	*J&C class*	*R&E class*
PKB326	63	Ribozymes	Branchiostoma floridae	complex	ABCDCADB	2	N	N	N	N	Y
PKB331	64	Ribozymes	Strongylocentrotus purpuratus	complex	ABCDCADB	2	N	N	N	N	Y
PKB330	64	Ribozymes	Strongylocentrotus purpuratus	complex	ABCDCADB	2	N	N	N	N	Y
PKB338	66	Ribozymes	Petromyzon marinus	complex	ABCDCADB	2	N	N	N	N	Y

Continued On Next Page

Table 6.4 – *Continued From Previous Page*

Name	Length	RNA Type	Organism	Pseudoknot Type	Pseudoknot pattern	Page No.	*L&P class*	*D&P class*	*A&U class*	*J&C class*	*R&E class*
PKB315	67	Ribozymes	Dog	complex	ABCDCADB	2	N	N	N	N	Y
PKB318	67	Ribozymes	Opossum	complex	ABCDCADB	2	N	N	N	N	Y
PKB319	67	Ribozymes	Mouse	complex	ABCDCADB	2	N	N	N	N	Y
PKB320	67	Ribozymes	Rabbit	complex	ABCDCADB	2	N	N	N	N	Y
PKB321	67	Ribozymes	Chimpanzee	complex	ABCDCADB	2	N	N	N	N	Y
PKB322	67	Ribozymes	Rat	complex	ABCDCADB	2	N	N	N	N	Y
PKB317	67	Ribozymes	Elephant	complex	ABCDCADB	2	N	N	N	N	Y
PKB316	67	Ribozymes	Homo sapiens	complex	ABCDCADB	2	N	N	N	N	Y
PKB340	67	Ribozymes	Invertebrate iridescent virus 6	complex	ABCDCADB	2	N	N	N	N	Y
PKB314	67	Ribozymes	Cow	complex	ABCDCADB	2	N	N	N	N	Y
PKB333	68	Ribozymes	Strongylocentrotus purpuratus	complex	ABCDCADB	2	N	N	N	N	Y
PKB332	68	Ribozymes	Strongylocentrotus purpuratus	complex	ABCDCADB	2	N	N	N	N	Y

Continued On Next Page

Table 6.4 – *Continued From Previous Page*

Name	Length	RNA Type	Organism	Pseudoknot Type	Pseudoknot pattern	Page No.	*L&P class*	*D&P class*	*A&U class*	*J&C class*	*R&E class*
PKB339	69	Ribozymes	Faecalibacterium prausnitzii	complex	ABCDCADB	2	N	N	N	N	Y
PKB341	76	Ribozymes	Diplonema papillatum	complex	ABCDCADB	2	N	N	N	N	Y
PKB334	77	Ribozymes	Strongylocentrotus purpuratus	complex	ABCDCADB	2	N	N	N	N	Y
PKB325	78	Ribozymes	Branchiostoma floridae	complex	ABCDCADB	2	N	N	N	N	Y
PKB328	81	Ribozymes	Anopheles gambiae	complex	ABCDCADB	2	N	N	N	N	Y
PKB327	82	Ribozymes	Anopheles gambiae	complex	ABCDCADB	2	N	N	N	N	Y
PKB329	82	Ribozymes	Anopheles gambiae	complex	ABCDCADB	2	N	N	N	N	Y
PKB342	88	Ribozymes	Trichoderma atroviride	complex	ABCDCADB	2	N	N	N	N	Y
PKB75	88	Ribozymes	Hepatitis Delta Virus	complex	ABCDCADB	2	N	N	N	N	Y

Continued On Next Page

Table 6.4 – *Continued From Previous Page*

Name	Length	RNA Type	Organism	Pseudoknot Type	Pseudoknot pattern	Page No.	*L&P class*	*D&P class*	*A&U class*	*J&C class*	*R&E class*
PKB335	104	Ribozymes	Caenorhabditis japonica	complex	ABCDCADB	2	N	N	N	N	Y
PKB337	106	Ribozymes	Pristionchus pacificus	complex	ABCDCADB	2	N	N	N	N	Y
PKB336	106	Ribozymes	Pristionchus pacificus	complex	ABCDCADB	2	N	N	N	N	Y
PKB356	140	Ribozymes	Drosophila ananassae	complex	ABCDCADB	2	N	N	N	N	Y
PKB355	150	Ribozymes	Drosophila yakuba	complex	ABCDCADB	2	N	N	N	N	Y
PKB357	160	Ribozymes	Drosophila pseudoobscura	complex	ABCDCADB	2	N	N	N	N	Y
PKB323	180	Ribozymes	Anopheles gambiae	complex	ABCDCADB	2	N	N	N	N	Y
PKB324	181	Ribozymes	Anopheles gambiae	complex	ABCDCADB	2	N	N	N	N	Y
PKB358	190	Ribozymes	Drosophila falleni	complex	ABCDCADB	2	N	N	N	N	Y

Continued On Next Page

Table 6.4 – *Continued From Previous Page*

Name	Length	RNA Type	Organism	Pseudoknot Type	Pseudoknot pattern	Page No.	*L&P class*	*D&P class*	*A&U class*	*J&C class*	*R&E class*
PKB354	190	Ribozymes	Drosophila simulans	complex	ABCDCADB	2	N	N	N	N	Y
3RKF_A	67	Guanine Riboswitch		complex	ABCBDADECE	2	N	N	N	N	Y
3IVN_B	69	Adenosine Riboswitch	Bacillus subtilis	complex	ABCBDADECE	2	N	N	N	N	Y
3LA5_A	71	Adenosine Riboswitch	Bacillus subtilis	complex	ABCBDADECE	2	N	N	N	N	Y
4FRN_A	102	Cobalamin riboswitch aptamer domain	Marine metagenome	complex	ABCDCEBEAFDF	2	N	N	N	N	Y
PKB71	108	mRNA	Escherichia coli	complex	ABCABC	3	N	N	N	N	Y
3ZEX_B	1465	28S rRNA	Trypanosoma brucei	complex	others	3	N	N	N	N	N
3J20_2	1495	16S rRNA	Pyrococcus furiosus DSM 3638	complex	others	3	N	N	N	N	Y

Continued On Next Page

Table 6.4 – *Continued From Previous Page*

Name	Length	RNA Type	Organism	Pseudoknot Type	Pseudoknot pattern	Page No.	*L&P class*	*D&P class*	*A&U class*	*J&C class*	*R&E class*
2WDL_A	2807	23S rRNA	Thermus thermophilus HB8	complex	others	4	N	N	N	N	N
3KIY_A	2848		Thermus thermophilus HB8	complex	others	4	N	N	N	N	N

6.1.4 Complex Pseudoknots with Page Number ≥ 3

If we focus on the most complex pseudoknots, we may wonder how complicated the crossing interactions are. In fact, in the purpose of illustrating the complex pseudoknots with page number ≥ 3, the schematic figures of the crossing interactions representing the conformation are provided.

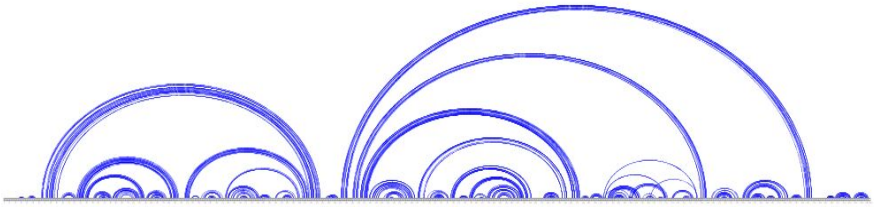
The pseudotrefoil *PKB71*, with the pseudoknot pattern of *ABCABC*, is quite easy to be understood as its page number of 3. The stems *AA*, *BB* and *CC* cross mutually, which requires exactly three pages to decompose it into the union of pseudoknot-free substructures.

On the other hand, 3ZEX_B, 3J20_2, 2WDL_A and 3KIY_A, the 4 complex pseudoknots display much more complicated conformations. Specifically, the schematic figures for demonstrating each complex pseudoknot are composed of the following four subfigures:

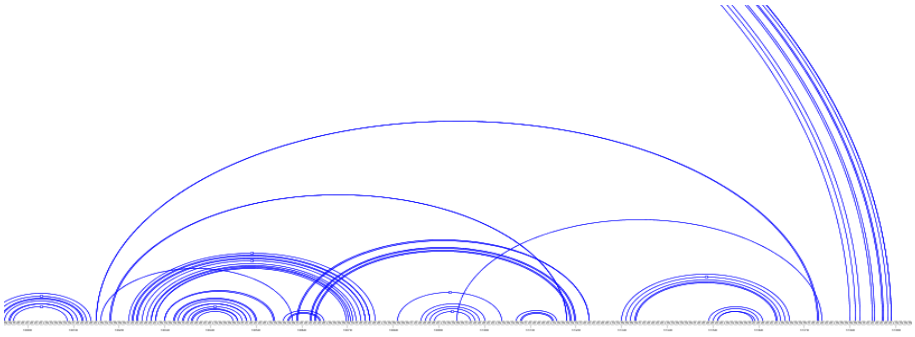
- One screen-shot of the global structure of the pseudoknot, visualized by VARNA.
- One screen-shot of the *dominant* local structure of the pseudoknot,visualized by VARNA. The notion of the *dominant* local structure used here declares the smallest substructure which has the same page number as the global conformation.
- The corresponding RNA shadow of the dominant local region.
- The coloring notation of the RNA shadow, corresponding to a decomposition of the pseudoknotted conformation with colors such that the nested base pairs are represented in an unique color.

6.1.5 3ZEX_B

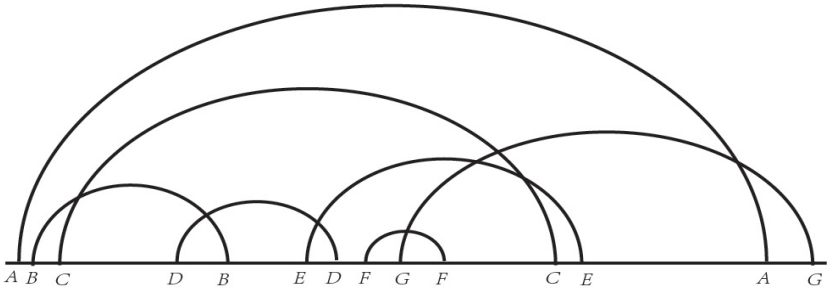
Figure 6.1(a) shows the global conformation of 3ZEX_B, while Figure 6.1(b) shows the dominant local part, from the 1015 base to the 1174 base. And Figure 6.1(c) corresponds to the shadow of the current dominant region, from which we may bear in mind how complicated the crossing interactions that 3ZEX_B has.



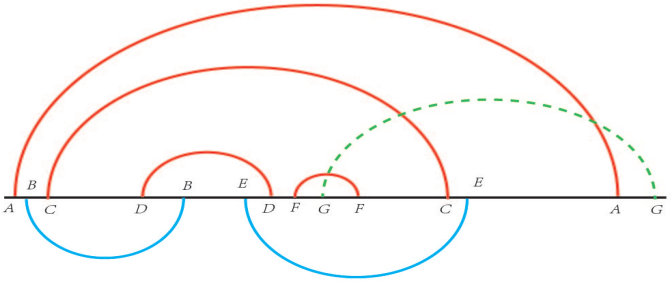
(a) The global structure of 3ZEX_B.



(b) The dominant local part of 3ZEX_B, 1015nt-1174nt.



(c) The corresponding shadow, 1015nt-1174nt.



(d) The coloring notation of the shadow, 1015nt-1174nt.

Figure 6.1: The schematic figures of 3ZEX_B.

We investigate the page number of 3ZEX_B by decomposing it into a union of nested base pairs in the context of RNA shadows, such that each set of nested base pairs are colored with an unique color.

Figure 6.1(d) shows that the decomposition of the shadow of 3ZEX_B requires three colors. Particularly, the base pair *GG* crosses both the base pairs represented in the upper semi-plane in red and the ones represented in the lower semi-plane in blue. It suggests that the base pair *GG* should be represented in a third semi-plane which is marked in the dashed line and colored with a third color.

The pseudoknot pattern of the dominant region of 3ZEX_B is *ABCDBED-FGFCEAG*. But since the global conformation still contains some local subpseudoknots which are nested inside the unpaired loops elsewhere, the pseudoknot pattern of 3ZEX_B is assigned as *others* by this benchmark.

6.1.6 3J20_2

Figure 6.2 shows the three subfigures of 3J20_2, where the local dominant region starts from the 523 base to the 1484 base.

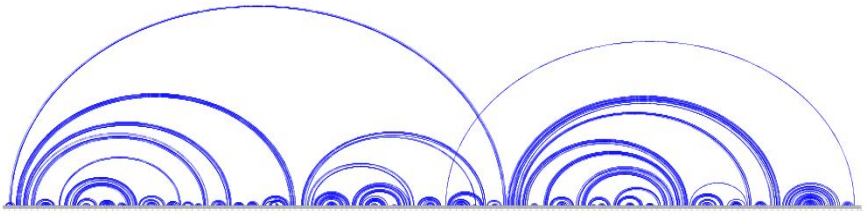
3J20_2 has relatively simple crossing interactions, compared to the 3ZEX_B. As shown in Figure 6.2(c), 3J20_2 has pseudotrefoil-like crossing interactions with the pattern of *ABCABC*, accompanied by another base pair *DD*. The decomposition of the dominant region of 3J20_2 in colors is shown in Figure 6.2(d), suggesting 3J20_2 a page number of 3.

The pseudoknot pattern of the dominant region of 3J20_2 is *ABCDADBC*. But the global pseudoknot pattern of 3J20_2 is assigned as *others* by this benchmark under the same consideration as 3ZEX_B.

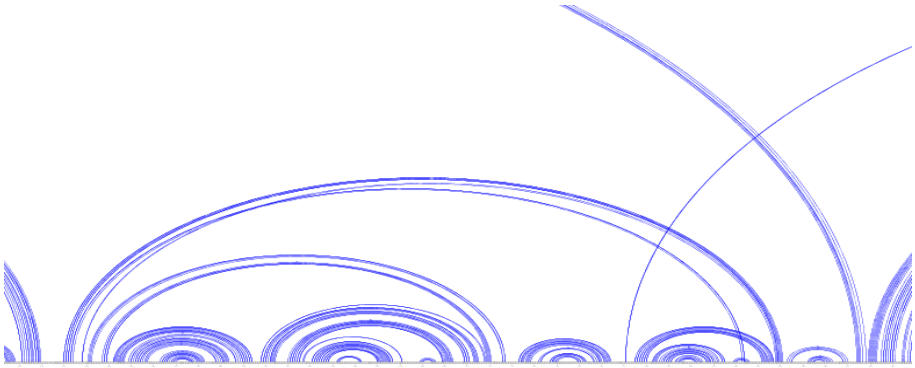
6.1.7 2WDL_A

Figure 6.3(a) shows the global conformation of 2WDL_A, Figure 6.3(b) shows the dominant local part, from the 429 base to the 2617 base, and Figure 6.3(c) shows the shadow of the dominant region.

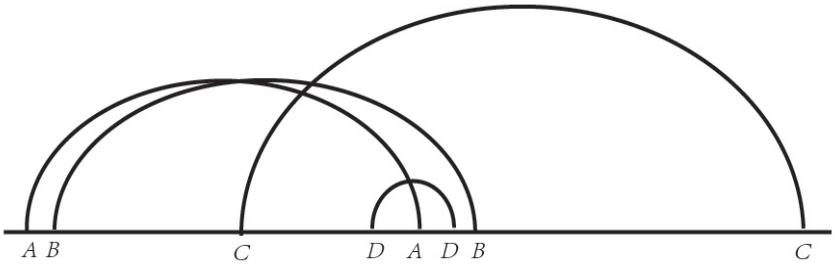
The decomposition of the dominant region of 2WDL_A in colors is shown in Figure 6.3(d).



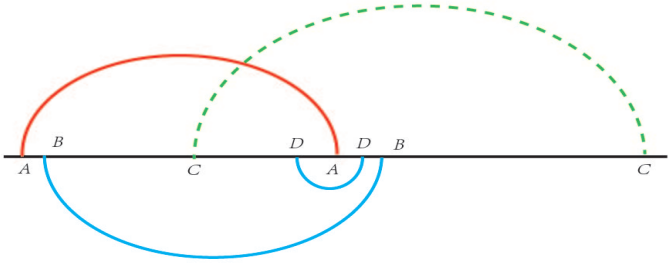
(a) The global structure of 3J20_2.



(b) The dominant local part of 3J20_2, 523nt-1484nt.

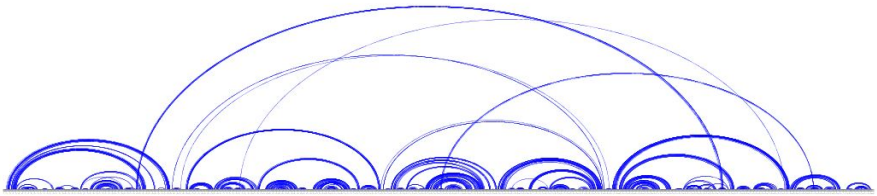


(c) The corresponding shadow, 523nt-1484nt.

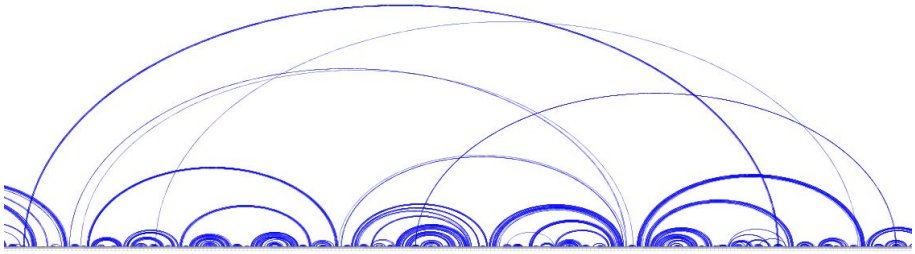


(d) The coloring notation of the shadow, 523nt-1484nt.

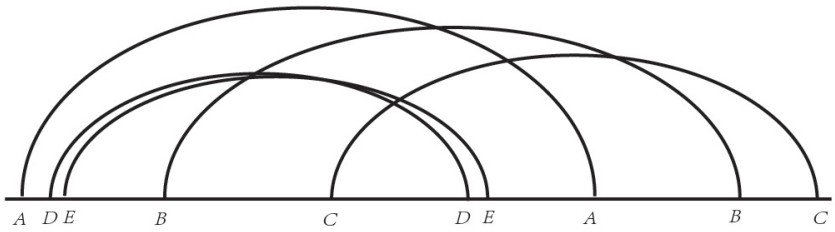
Figure 6.2: The schematic figures of 3J20_2.



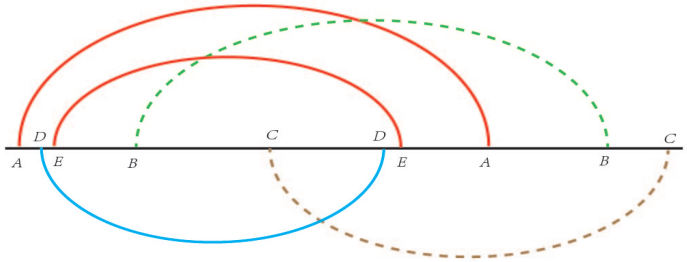
(a) The global structure of 2WDL_A.



(b) The dominant local part of 2WDL_A, 429nt-2617nt.



(c) The corresponding shadow, 429nt-2617nt.



(d) The coloring notation of the shadow, 429nt-2617nt.

Figure 6.3: The schematic figures of 2WDL_A.

Particularly, the base pair *BB* crosses both the base pairs *AA* and *EE* represented in the upper semi-plane in red and *DD* represented in the lower semi-plane in blue. It suggests that the base pair *BB* should be represented in a third semi-plane with a third color. Further, the base pair *CC* crosses all the bases pairs represented in the last three semi-planes, suggesting its representation in a fourth semi-plane with a fourth color, which is shown in brown dashed line in Figure 6.3(d).

As a consequence, the page number of the dominant region of 2WDL_A is 4, so is the global conformation. The pseudoknot pattern of the dominant region of 2WDL_A is *ADEBCDEABC*, and the global pseudoknot pattern of 2WDL_A is assigned as *others* by this benchmark, as some local substructures are located outside the dominant region.

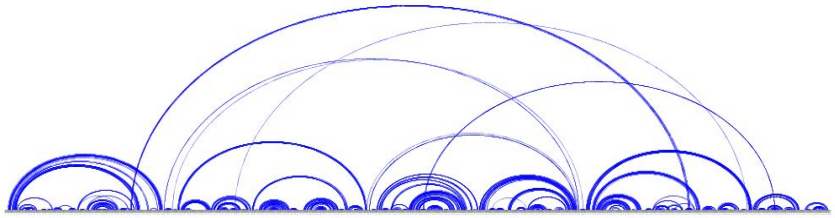
6.1.8 3KIY_A

3KIY_A is another example which has a page number of 4 in this benchmark. Similarly, Figure 6.4(a) shows the global conformation of 3KIY_A, Figure 6.4(b) shows the local dominant part, from the 434 base to the 2658 base, and Figure 6.4(c) shows the shadow of the dominant region of 3KIY_A. The decomposition of the dominant region of 3KIY_A in colors is shown in Figure 6.4(d).

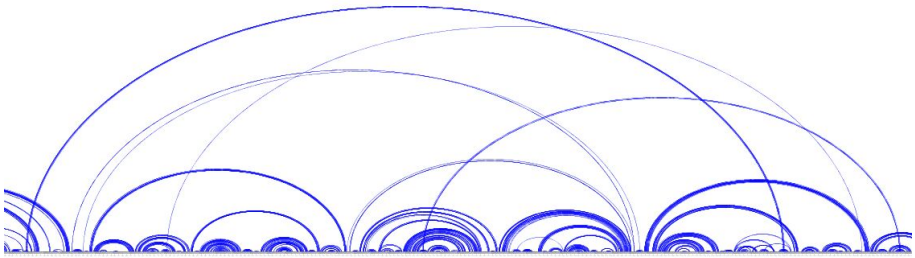
In fact, as shown in Figure 6.4(c), 3KIY_A has the same shadow for the dominant region as 2WDL_A. The same pseudotrefoil-like crossing interactions with the pattern of *DEBCDEBC* contributes both the dominant structure and global conformation of 3KIY_A a page number of 4.

6.2 Prediction of the Pseudoknots

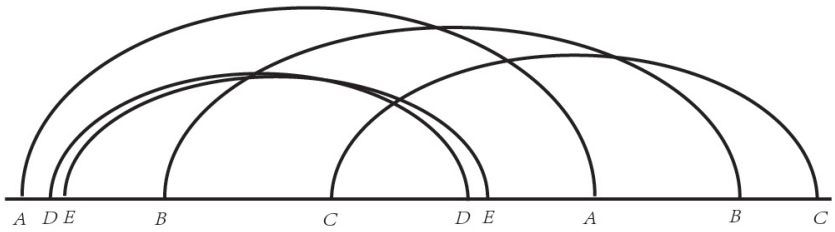
A series of comparisons are carried out which aims to compare the performance of predicting pseudoknots by each method. In practice, the performance of predictions is assessed based on the entire dataset, the shared set of sequences, and hierarchical subsets of pseudoknots which are divided by levels of complexity measurement of the pseudoknots, and the length, organism and RNA type of the sequences.



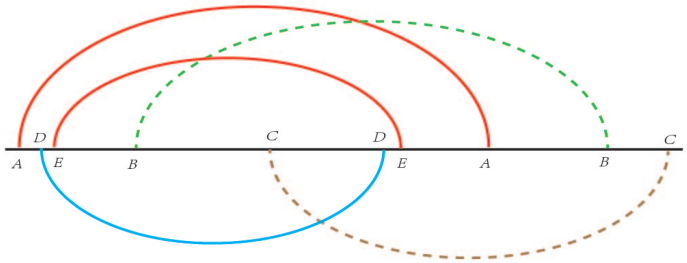
(a) The global structure of 3KIY_A.



(b) The dominant local part of 3KIY_A, 434nt-2658nt.



(c) The corresponding shadow, 434nt-2658nt.



(d) The coloring notation of the shadow, 434nt-2658nt.

Figure 6.4: The schematic figures of 3KIY_A.

A comprehensive classification of the pseudoknots and respective sizes is shown in Table 6.5, as well as the numeric values of pseudoknots that each method can handle. The ability of handling or predicting a sequence means that the method can return for the input sequence a secondary structure or several ones, with or without pseudoknots, but offers no guarantee of the quality of the prediction.

Particularly, the input length thresholds in Table 6.5 are given by the longest sequence that the method can handle and the shortest one it can not. For example, the longest sequence that CyloFold can handle is 412 nucleotides, and the shortest one that CyloFold fails to predict a secondary structure is 920 nucleotides. Consequently, the input length threshold of CyloFold is longer than 412 nucleotides but shorter than 920 nucleotides, the same to that of HotKnots-dp, HotKnots-re and vsfold5.

In addition, the unassigned values in the *Organism* and *RNA Type* classification of the sequences are considered as well, marked with a value *Unknown* in Table 6.5.

Meanwhile, as mentioned in Section 5.1, a subset of pseudoknots containing the sequences which can be handled by all the benchmarking methods is considered, with the consideration of comparing the predictions of all the methods with a consistent set of sequences. In fact, the threshold of choosing sequences for this shared subset depends on MC-Fold, as it has the most restricted requirement on the length of the input sequence.

Finally, out of 414 sequences, 387 sequences that MC-Fold can handle compose the *shared set*. And the *non-shared*, or *missing* 27 sequences left from the entire dataset compose the *missing set*.

Table 6.5: The numeric value of the predictions.

Attribute	Value	Size	CyloFold	DotKnot	DotKnot-K	HotKnots-cc	HotKnots-dp	HotKnots-re	IPknot	MC-Fold	McGenus	McQFold	pKiss	vsfold5	pknotsRG-MFE	pknotsRG-FE	pknots
Input Length Threshold (nt)			$\geq 412^{1}$ < 920	≥ 3170	≥ 3170	≥ 920 < 927	≥ 412 < 920	≥ 412 < 920	≥ 3170	≥ 158 < 160	≥ 412 < 920	≥ 1248 < 1465	≥ 190 < 207	≥ 412 < 920	≥ 1248 < 1465	≥ 1248 < 1465	≥ 212 < 219
All	PseudoBase	367	365	367	367	366	365	365	367	350	365	367	355	365	367	367	357
			99.5%	100%	100%	99.7%	99.5%	99.5%	100%	95.4%	99.5%	100%	96.7%	99.5%	100%	100%	97.3%
	PDB	47	41	47	47	41	41	41	47	37	41	42	39	41	42	42	39
			87.2%	100%	100%	87.2%	87.2%	87.2%	100%	78.7%	87.2%	89.4%	83.0%	87.2%	89.4%	89.4%	83.0%
Length(nt)	≤ 100	345	345	345	345	345	345	345	345	345	345	345	345	345	345	345	345
			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	$101 - 200$	49	49	49	49	49	49	49	49	42	49	49	49	49	49	49	49
			100%	100%	100%	100%	100%	100%	100%	85.7%	100%	100%	100%	100%	100%	100%	100%
	$201 - 300$	6	6	6	6	6	6	6	6	0	6	6	0	6	6	6	2
			100%	100%	100%	100%	100%	100%	100%	0%	100%	100%	0%	100%	100%	100%	33.3%
	$301 - 400$	5	5	5	5	5	5	5	5	0	5	5	0	5	5	5	0
			100%	100%	100%	100%	100%	100%	100%	0%	100%	100%	0%	100%	100%	100%	0%
	$401 - 500$	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0
			100%	100%	100%	100%	100%	100%	100%	0%	100%	100%	0%	100%	100%	100%	0%
	$501 - 1000$	2	0	2	2	1	0	0	2	0	0	2	0	0	2	2	0
			0%	100%	100%	50.0%	0%	0%	100%	0%	0%	100%	0%	0%	100%	100%	0%

Continued On Next Page

Table 6.5 – *Continued From Previous Page*

Attribute	Value	Size	CyloFold	DotKnot	DotKnot-K	HotKnots-cc	HotKnots-dp	HotKnots-re	IPknot	MC-Fold	McGenus	McQFold	pKiss	vsfold5	pknotsRG-MST	pknotsRG-FST	pknots
	≥ 1001	6	0	6	6	0	0	0	6	0	0	1	0	0	1	1	0
			0%	100%	100%	0%	0%	0%	100%	0%	0%	16.7%	0%	0%	16.7%	16.7%	0%
Organism	Eukaryote	68	66	68	68	66	66	66	68	59	66	66	65	66	66	66	65
			97.1%	100%	100%	97.1%	97.1%	97.1%	100%	86.8%	97.1%	97.1%	95.6%	97.1%	97.1%	97.1%	95.6%
	Prokaryote	52	47	52	52	48	47	47	52	40	47	49	41	47	49	49	42
			90.4%	100%	100%	92.3%	90.4%	90.4%	100%	76.9%	90.4%	94.2%	78.8%	90.4%	94.2%	94.2%	80.8%
	Virus	272	271	272	272	271	271	271	272	266	271	272	266	271	272	272	267
			99.6%	100%	100%	99.6%	99.6%	99.6%	100%	97.8%	99.6%	100%	97.8%	99.6%	100%	100%	98.2%
	Unknown	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Pknot Type	H-type	341	339	341	341	340	339	339	341	330	339	341	331	339	341	341	332
			99.4%	100%	100%	99.7%	99.4%	99.4%	100%	96.8%	99.4%	100%	97.1%	99.4%	100%	100%	97.4%
	Kissing	25	25	25	25	25	25	25	25	22	25	25	23	25	25	25	24
			100%	100%	100%	100%	100%	100%	100%	88.0%	100%	100%	92.0%	100%	100%	100%	96.0%
	Complex	44	40	44	44	40	40	40	44	35	40	40	40	40	40	40	40
			90.9%	100%	100%	90.9%	90.9%	90.9%	100%	79.5%	90.9%	90.9%	90.9%	90.9%	90.9%	90.9%	90.9%
	Recursive	4	2	4	4	2	2	2	4	0	2	3	0	2	3	3	0
			50.0%	100%	100%	50.0%	50.0%	50.0%	100%	0%	50.0%	75.0%	0%	50.0%	75.0%	75.0%	0%

Continued On Next Page

Table 6.5 – *Continued From Previous Page*

Attribute	Value	Size	CyloFold	DotKnot	DotKnot-K	HotKnots-cc	HotKnots-dp	HotKnots-re	IPknot	MC-Fold	McGenus	McQFold	pKiss	vsfold5	pknotsRG-MFE	pknotsRG-F	pknots
Page No.	2	409	405	409	409	406	405	405	409	386	405	408	393	405	408	408	395
			99.0%	100%	100%	99.3%	99.0%	99.0%	100%	94.4%	99.0%	99.8%	96.1%	99.0%	99.8%	99.8%	96.6%
	3	3	1	3	3	1	1	1	3	1	1	1	1	1	1	1	1
			33.3%	100%	100%	33.3%	33.3%	33.3%	100%	33.3%	33.3%	33.3%	33.3%	33.3%	33.3%	33.3%	33.3%
	4	2	0	2	2	0	0	0	2	0	0	0	0	0	0	0	0
			0%	100%	100%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
RNA Type	Aptamers	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	mRNA	17	17	17	17	17	17	17	17	16	17	17	16	17	17	17	16
			100%	100%	100%	100%	100%	100%	100%	94.1%	100%	100%	94.1%	100%	100%	100%	94.1%
	tRNA	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	tmRNA	12	12	12	12	12	12	12	12	10	12	12	10	12	12	12	10
			100%	100%	100%	100%	100%	100%	100%	83.3%	100%	100%	83.3%	100%	100%	100%	83.3%
	rRNA	18	12	18	18	13	12	12	18	10	12	14	11	12	14	14	11
			66.7%	100%	100%	72.2%	66.7%	66.7%	100%	55.6%	66.7%	77.8%	61.1%	66.7%	77.8%	77.8%	61.1%
	Riboswitch	12	12	12	12	12	12	12	12	11	12	12	12	12	12	12	12
			100%	100%	100%	100%	100%	100%	100%	91.7%	100%	100%	100%	100%	100%	100%	100%

Continued On Next Page

Table 6.5 – *Continued From Previous Page*

Attribute	Value	Size	CyloFold	DotKnot	DotKnot-K	HotKnots-cc	HotKnots-dp	HotKnots-re	IPknot	MC-Fold	McGenus	McQFold	pKiss	vsfold5	pknotsRG-MC	pknotsRG-F	pknots
RNA Type	Ribozymes	45	45	45	45	45	45	45	45	37	45	45	42	45	45	45	43
			100%	100%	100%	100%	100%	100%	100%	82.2%	100%	100%	93.3%	100%	100%	100%	95.6%
	Others	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Vr. 3 UTR	103	103	103	103	103	103	103	103	103	103	103	103	103	103	103	103
			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Vr. 5 UTR	32	32	32	32	32	32	32	32	29	32	32	29	32	32	32	30
			100%	100%	100%	100%	100%	100%	100%	90.6%	100%	100%	90.6%	100%	100%	100%	93.8%
	Frameshifting	34	34	34	34	34	34	34	34	33	34	34	33	34	34	34	33
			100%	100%	100%	100%	100%	100%	100%	97.1%	100%	100%	97.1%	100%	100%	100%	97.1%
	Vr. Readthrough	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Vr. tRNA-like	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58
			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Vr. Others	32	31	32	32	31	31	31	32	30	31	32	30	31	32	32	30
			96.9%	100%	100%	96.9%	96.9%	96.9%	100%	93.8%	96.9%	100%	93.8%	96.9%	100%	100%	93.8%
	Unknown	8	7	8	8	7	7	7	8	7	7	7	7	7	7	7	7
			87.5%	100%	100%	87.5%	87.5%	87.5%	100%	87.5%	87.5%	87.5%	87.5%	87.5%	87.5%	87.5%	87.5%

[1]The manual of CyloFold mentions that the restriction on the length of input sequence is shorter than 550 nucleotides.

6.2.1 Average Performance

Based on the 387 shared sequences, and the subsets divided hierarchically, as shown in Tables 6.2 and 6.5, the density diagrams of the average sensitivity, PPV and the MCC values obtained by each method are shown in Figures 6.5, 6.6 and 6.7 respectively, where different classifications of the sequences are separated with empty rows in blue. Typically, the three evaluation parameters are calculated via the Equations 5.1, 5.2 and 5.3 defined in Section 5.5.

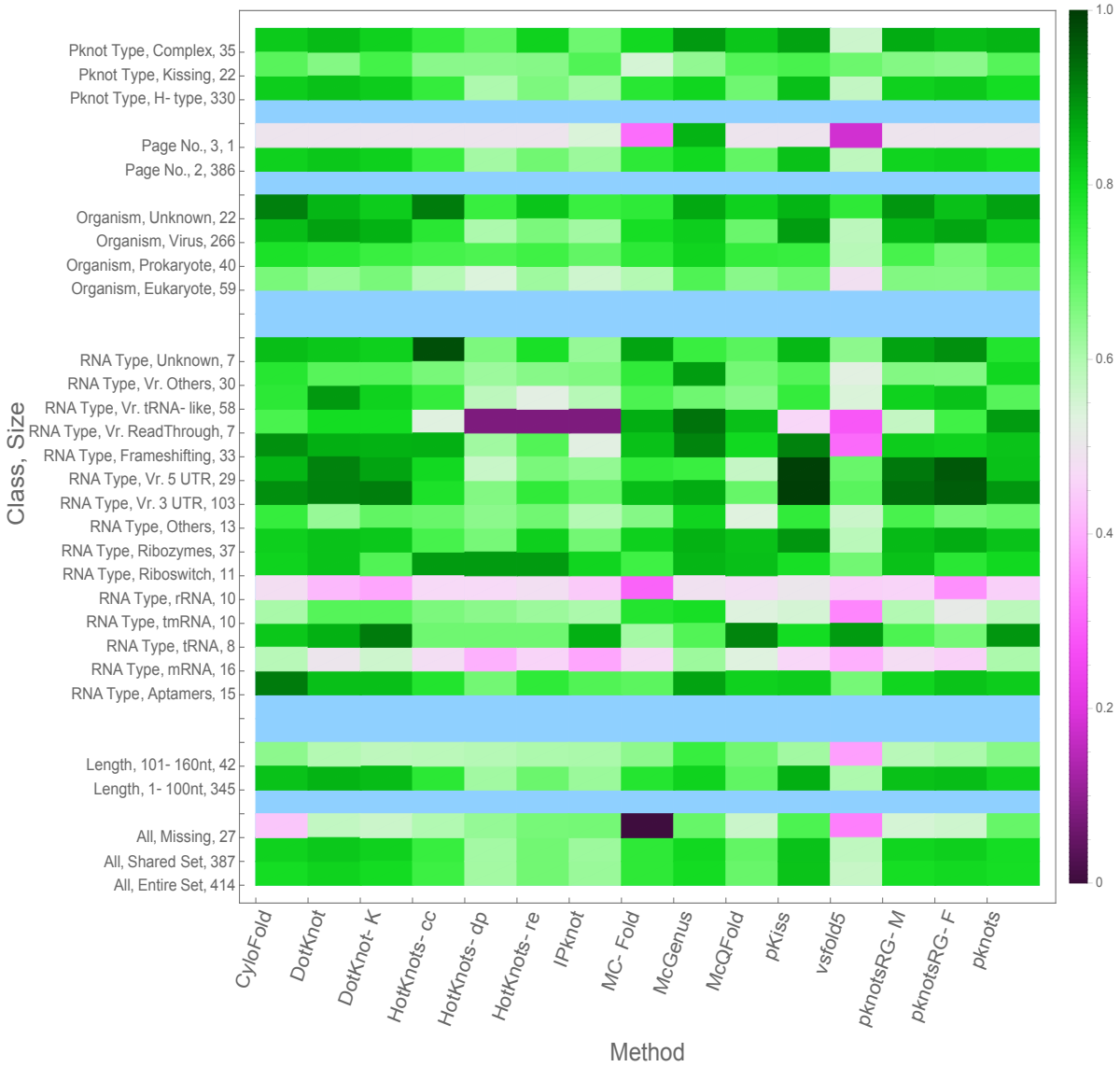


Figure 6.5: The density diagram of the sensitivity of the predictions.

In detail, the x-axis is labeled with the methods, where the twelve heuristic methods are listed before the three exact methods. And the y-axis is labeled with the different classes and corresponding number of sequences inside.

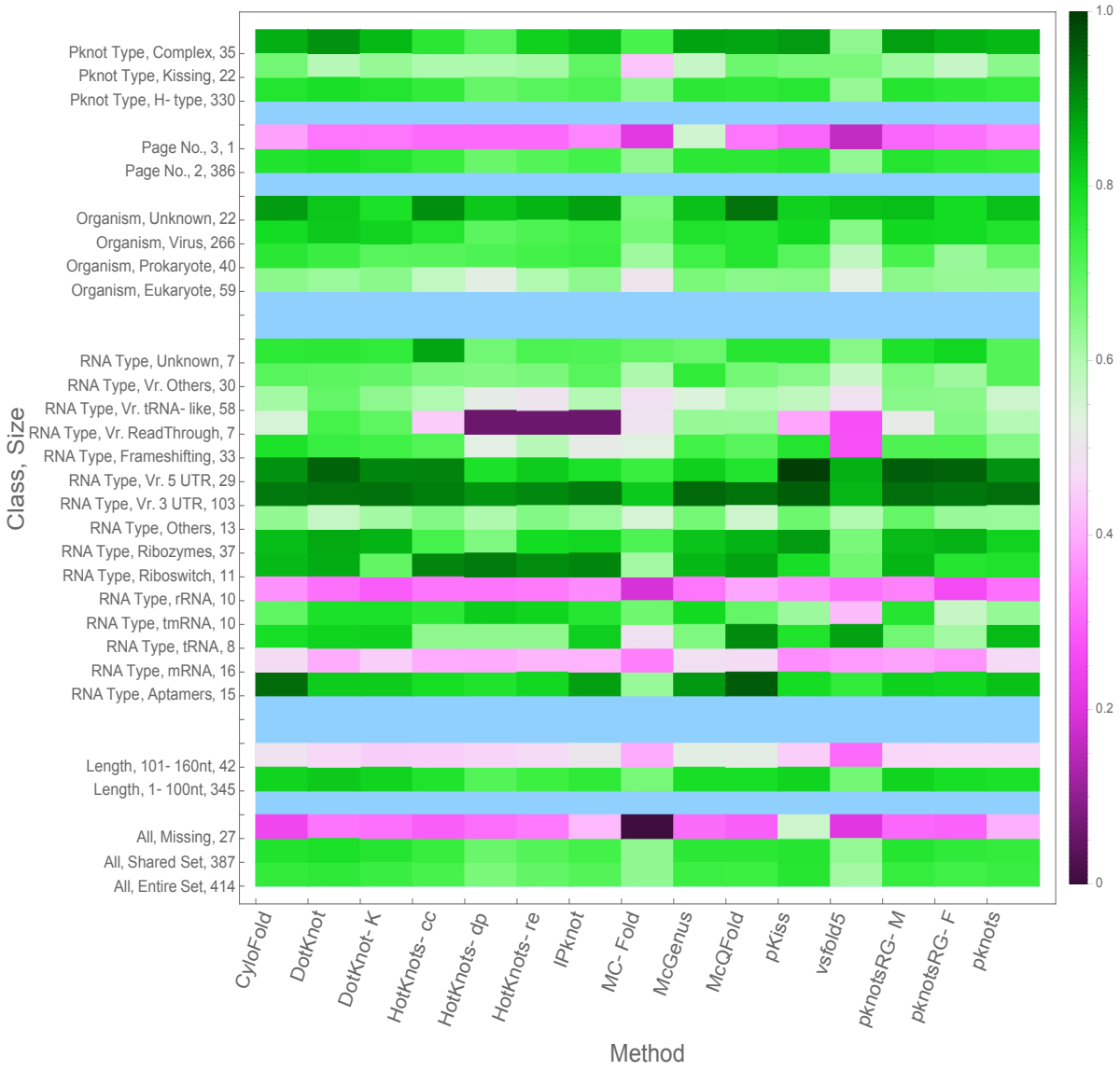


Figure 6.6: The density diagram of the PPV of the predictions.

Particularly, each prediction by one method for one class of sequences correspond to a rectangular box in each figure, and the evaluation values are reflected by the density of the colors in this box. The darker the green filled in the box is, the better the average prediction is, and the darker the fuchsia is, the worse the

average prediction is.

The detailed tables containing the evaluation values of the predictions, from which the three figures are generated, are provided in the Appendix D.

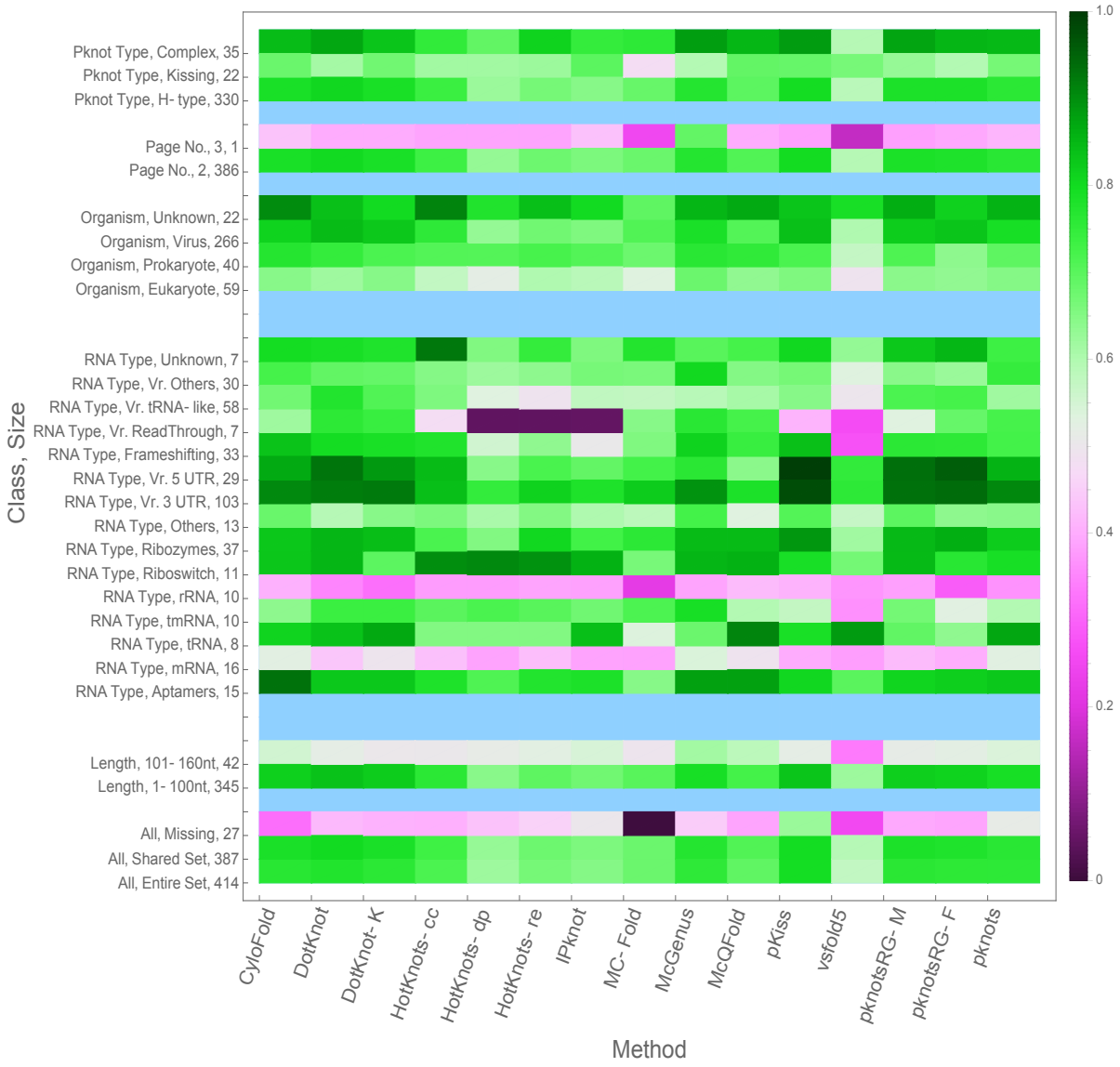


Figure 6.7: The density diagram of the MCC of the predictions.

Particularly, we select a corresponding winner program which has obtained an optimal prediction on average for each class of sequences in Figures 6.5, 6.6 and 6.7, as concluded in Table 6.6.

Table 6.6: The winner program of the evaluation values.

Attribute	Value	Size	Sensitivity	PPV	MCC
All	Entire Set	414	pKiss	pKiss	pKiss
	Shared Set	387	pKiss	DotKnot	DotKnot
	Missing Set	27	pKiss	pKiss	pKiss
Length	1-100 nt	345	pKiss	DotKnot	DotKnot
	101-160 nt	42	McGenus	McGenus	McGenus
RNA Type	Aptamers	15	CyloFold	McQFold	CyloFold
	mRNA	16	McGenus	McGenus	McGenus
	tRNA	8	DotKnot-K	McQFold	McQFold
	tmRNA	10	McGenus	HotKnots-dp	McGenus
	rRNA	10	pKiss	McQFold	McQFold
	Riboswitch	11	HotKnots-dp	HotKnots-dp	HotKnots-dp
	Ribozymes	37	pKiss	pKiss	pKiss
	Others	13	McGenus	pknotsRG-M	McGenus
	Vr. 3 UTR	103	pKiss	pKiss	pKiss
	Vr. 5 UTR	29	pKiss	pKiss	pKiss
	Frameshifting	33	pKiss	CyloFold	pKiss
	Vr. ReadThrough	7	McGenus	DotKnot	McGenus
	Vr. tRNA-like	58	DotKnot	DotKnot	DotKnot
	Vr. Others	30	McGenus	McGenus	McGenus
	Unknown	7	HotKnots-cc	HotKnots-cc	HotKnots-cc
Organism	Eukaryote	59	McGenus	McGenus	McGenus
	Prokaryote	40	McGenus	McQFold	CyloFold
	Virus	266	pKiss	DotKnot	DotKnot
	Unknown	22	HotKnots-cc	McQFold	HotKnots-cc
Page No.	2	386	pKiss	DotKnot	DotKnot
	3	1	McGenus	McGenus	McGenus
Pseudoknot Type	H-type	330	pKiss	DotKnot	DotKnot
	Kissing	22	DotKnot-K	IPknot	IPknot
	Complex	35	McGenus	DotKnot	pKiss
The Program with the			pKiss	DotKnot	McGenus
Maximal Times of Being a Winner			12	8	8

Figure 6.8 counts the winner program of the three evaluation parameters in Table 6.6, from which we observe the best three winner programs intuitively, DotKnot, McGenus and pKiss. These three methods obtain 15, 23 and 24 times

of best evaluation values respectively, accumulated based on the sensitivity, PPV and MCC. We term them as the *specific* winner methods, which have the maximum number of times in achieving the averagely optimal performance on predicting a particular class.

Contrarily, it is obvious to observe that the programs, such as MC-Fold, vsfold5 have a bare preponderance in this competition, as they are not present in Table 6.6 and with no score in Figure 6.8 at all.

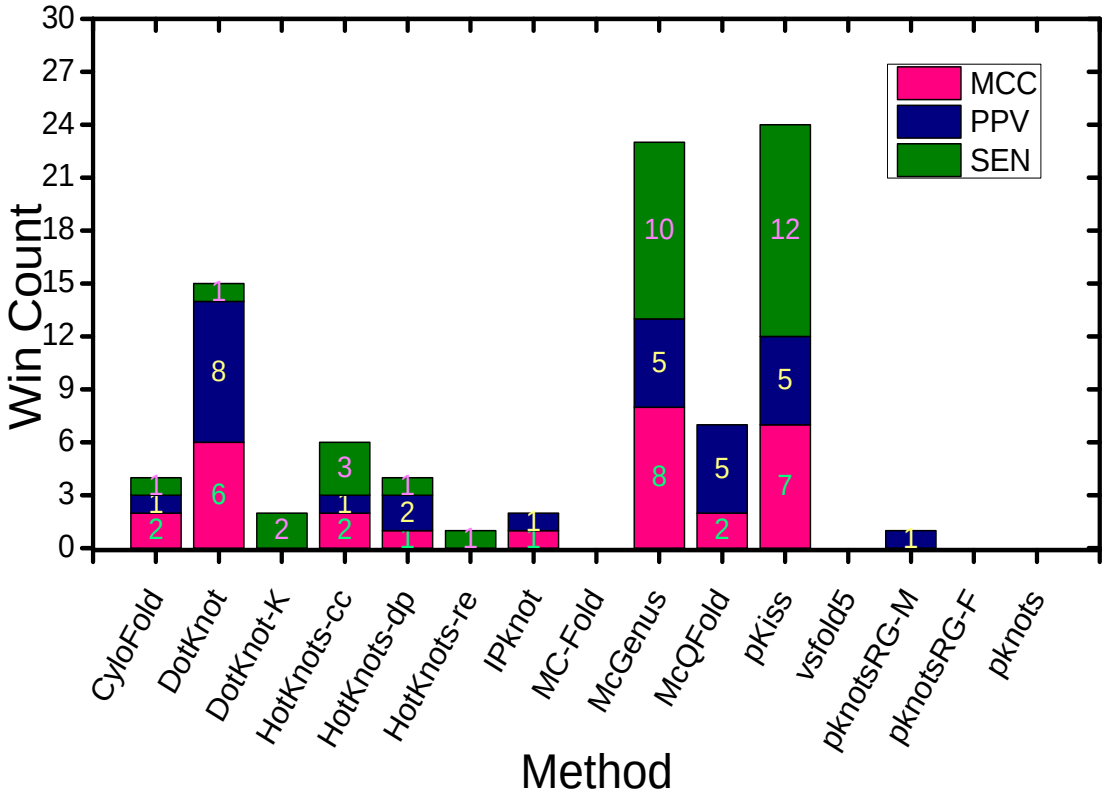


Figure 6.8: The win counts of each method.

Further, we may wonder how these specific winner methods perform on predicting other classes? Are they always good choices or just effective to some certain classes?

Figure 6.9 shows the global prediction performance of each method, where the line corresponds to the evaluation values ranging between the maximal and minimal ones that the certain method, labeled in the x-axis, has obtained, and

the solid point represents the arithmetic average of the evaluation values.

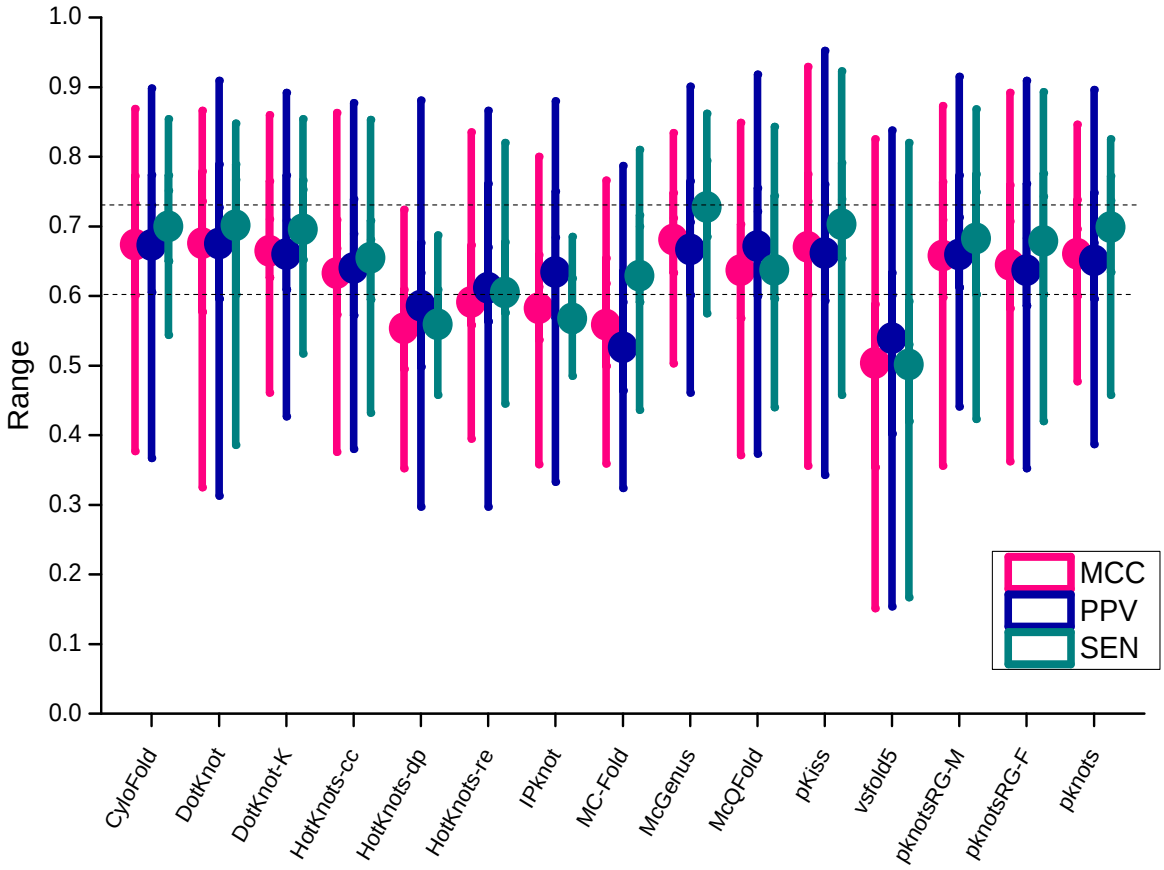


Figure 6.9: The global sensitivity, PPV and MCC of each prediction method.

We notice that the performance values of the majority of prediction methods aggregate between 0.6 and 0.75, which two are denoted by two dashed lines in Figure 6.9. The excluded five methods, HotKnots-dp, HotKnots-re, IPknot, MC-Fold and vsfold5 are not taken into account in the further comparisons, as some or all of their evaluation values are beneath 0.6.

But which of the other ten prediction methods are the best ones generally?

In the investigation of the answer, we consider to estimate a *consensus ranking* which suggests a single ranking that best 'agrees' with all the individual preference rankings of predicting the classes in Table 6.6. In practice, we employ the heuristic *BioConsert* in the *Median Ranking* web service [Brancotte et al., 2015] to perform the consensus ranking, which is a heuristic method designed for the biological data consensus ranking with *ties*. In this ranking model, some of ranking elements are

allowed to be aggregated which are considered as one group with a same ranking.

In detail, we have divided the 414 pseudoknots into 29 classes hierarchically in accordance to both the characteristic of sequences and the classification of pseudoknots, as shown in Figures 6.5, 6.6 and 6.7. And based on each class of sequences, there is an individual ranking of the prediction performance of the 15 methods, which is considered as one input of the consensus ranking.

Based on 29 inputs, the implementation of BioConsert have returned the consensus ranking of the prediction methods, with respect to the sensitivity, PPV and MCC respectively, as shown in Table 6.7.

Table 6.7: The consensus ranking of the prediction methods.

Rk.	Sensitivity	PPV	MCC	Rk.	Sensitivity	PPV	MCC
1	pKiss	DotKnot	DotKnot	9	MC-Fold	IPknot	pknotsRG-F
2	McGenus	CyloFold	pKiss	10	McQFold	pknotsRG-F	HotKnots-cc
3	DotKnot	McGenus	CyloFold	11	HotKnots-cc	HotKnots-cc	HotKnots-re
4	DotKnot-K	DotKnot-K	McGenus	12	HotKnots-re	HotKnots-re	IPknot
5	CyloFold	McQFold	DotKnot-K	13	IPknot	HotKnots-dp	MC-Fold
6	pknotsRG-M	pknotsRG-M	pknotsRG-M	14	HotKnots-dp	MC-Fold	HotKnots-dp
7	pknots	pKiss	pknots	15	vsfold5	vsfold5	vsfold5
8	pknotsRG-F	pknots	McQFold				

According to the Table 6.7, the best three prediction methods in the consensus ranking vary along with the three evaluation values, the sensitivity, PPV and MCC. But if we take the union of the top three ranked methods of each evaluation value, we have the following four ones: CyloFold, DotKnot, McGenus and pKiss. These four prediction methods are termed as the *global* winner methods, which have achieved the globally optimal performance on predicting all the classes.

Very interestingly, three of the global winner methods correspond to the specific winner methods, suggesting the union of both types of winner programs is comprised by the four global winner methods.

As a consequence, DotKnot, pKiss, CyloFold and McGenus are termed as the winner methods of this benchmark.

With respect to a comprehensive understanding of the relation between the 29 classes of sequences, as shown in Table 6.2, we plot the performance of the four

methods on predicting the functional families in the polar diagrams, as shown in Figures 6.10, 6.11 and 6.12. Particularly, the functional families and respective sizes are located along the circumference, and the radius correspond to the evaluation values ranging from 0 to 1, where the detailed sensitivity, PPV, and MCC of the predictions by the four winner programs are connected by lines.

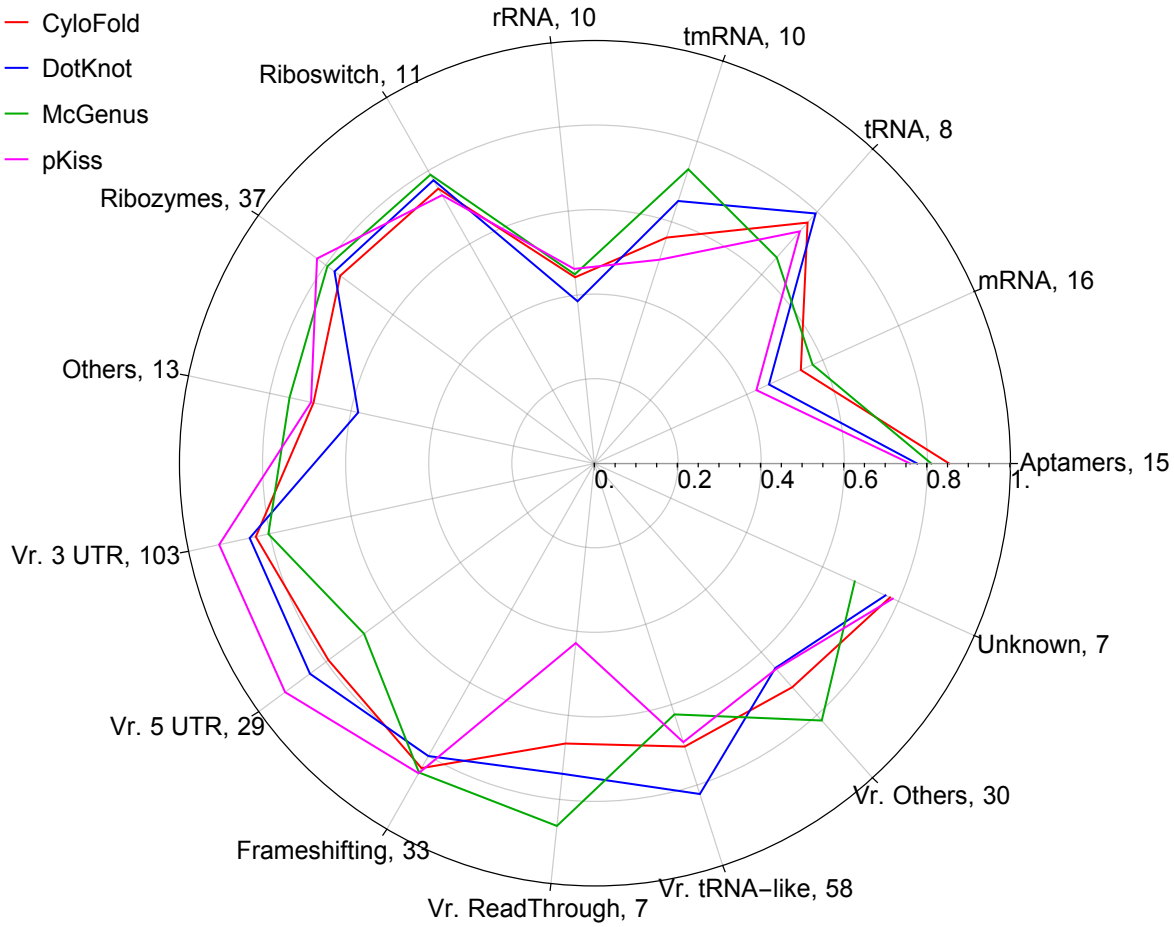


Figure 6.10: The sensitivity of predicting functional families by DotKnot, pKiss, CyloFold and McGenus.

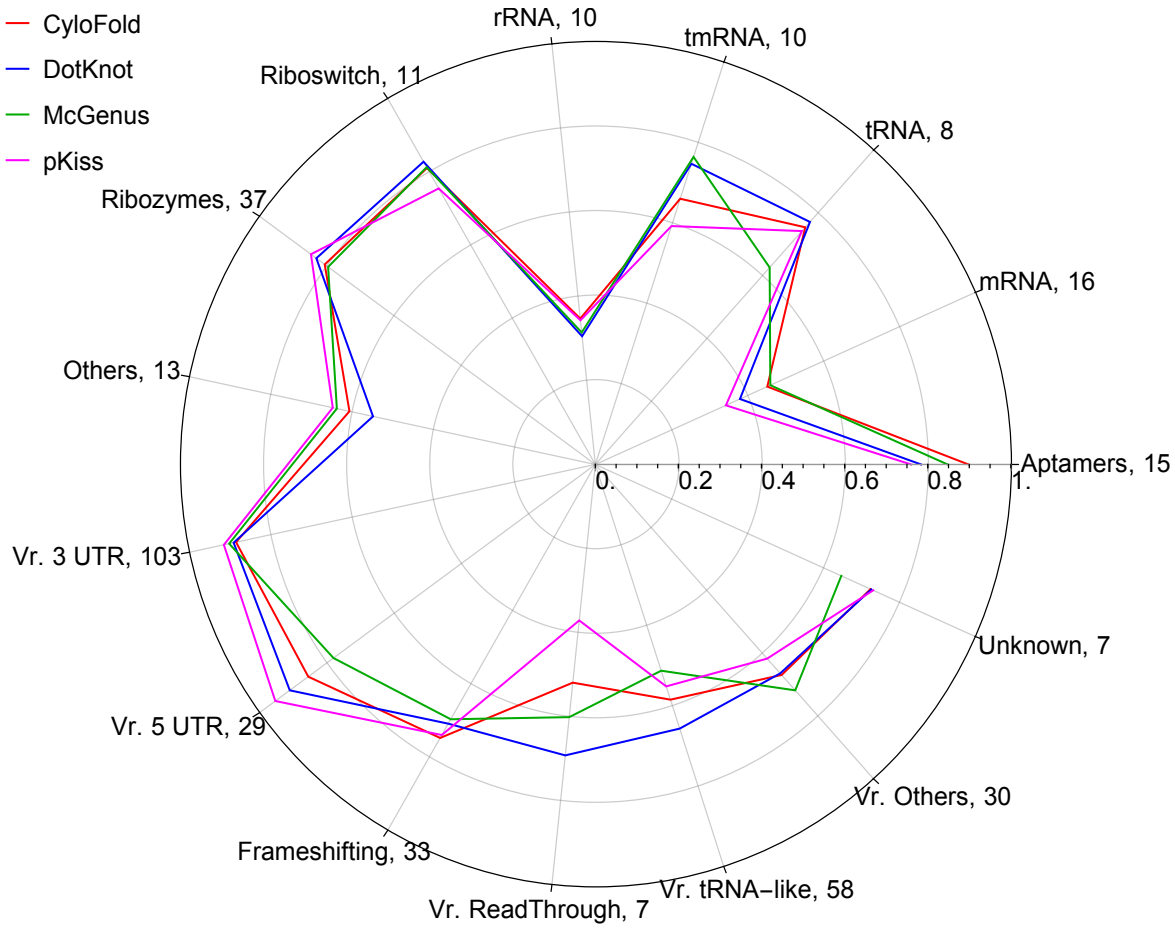


Figure 6.11: The PPV of predicting functional families by DotKnot, pKiss, CyloFold and McGenus.

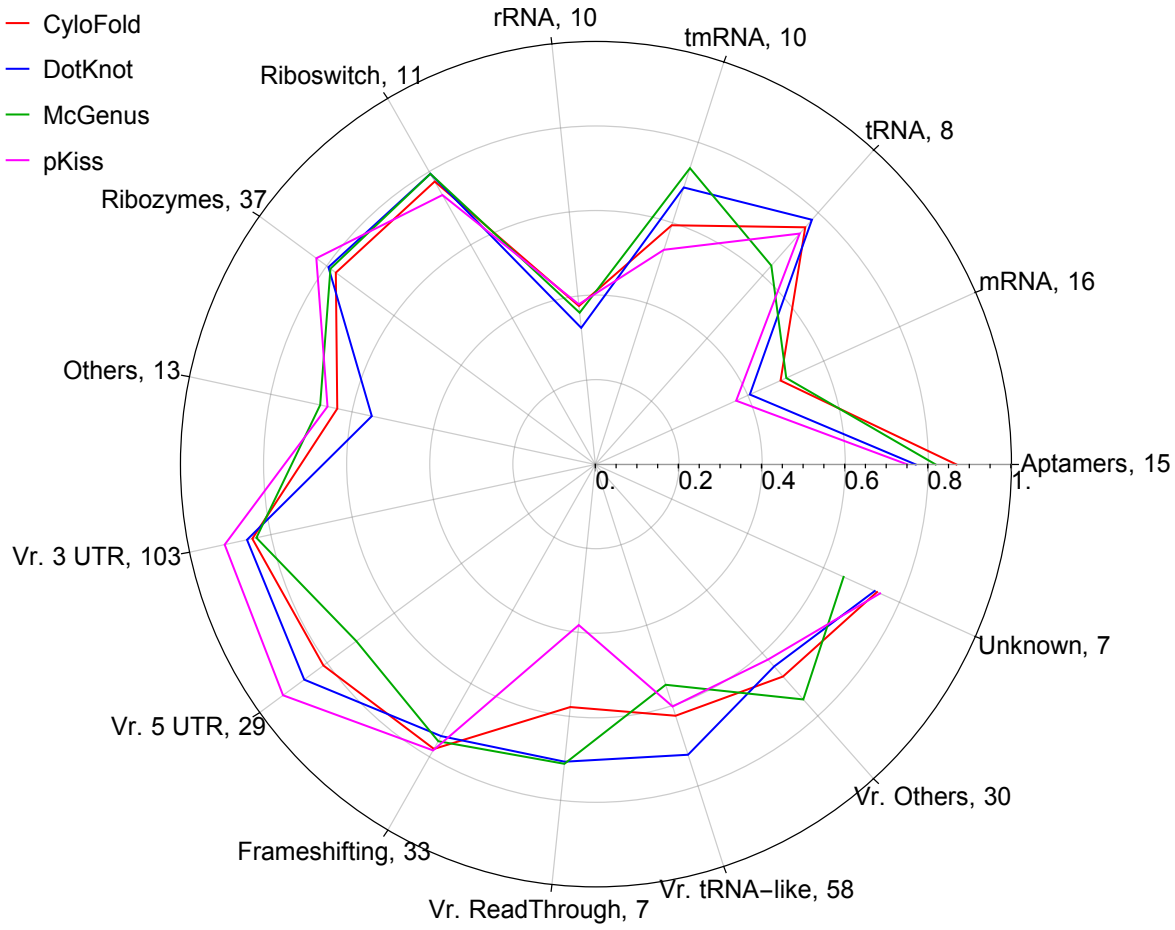
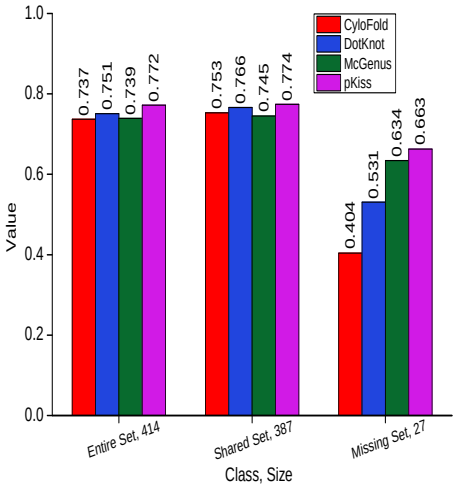
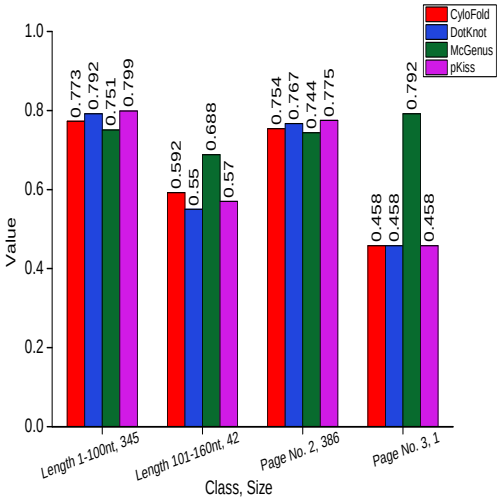


Figure 6.12: The MCC of predicting functional families by DotKnot, pKiss, CyloFold and McGenus.

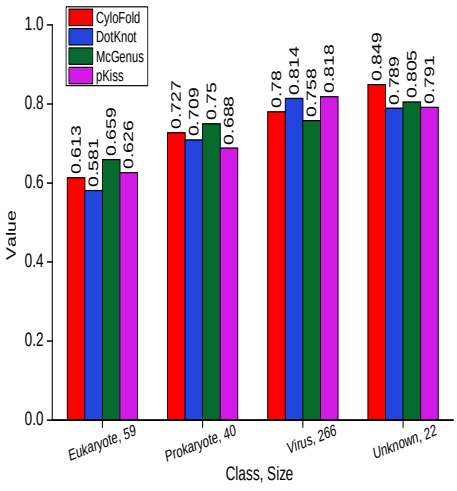
And we plot the other classifications of Table 6.2 in the histograms, as shown in Figures 6.13, 6.14 and 6.15. Particularly, the x-axis is label with the classes and their sizes, and the y-axis is label with the evaluation values ranging from 0 to 1.



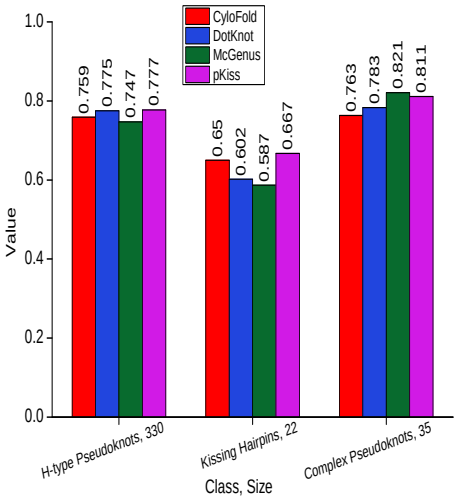
(a) Based on global groups.



(b) Based on the length and page no. of the sequences.

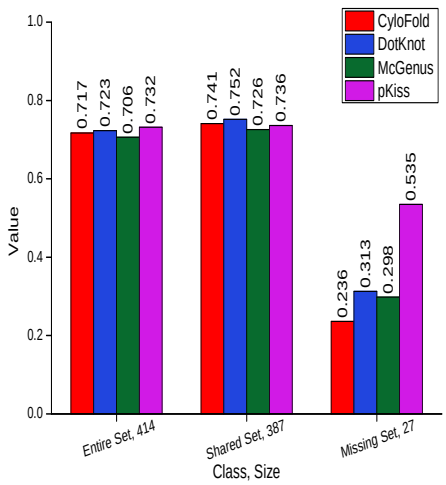


(c) Based on organisms.

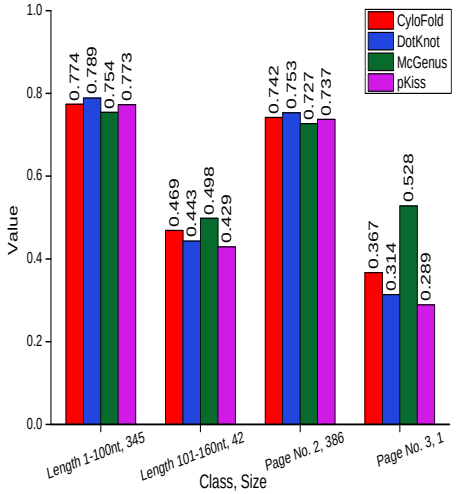


(d) Based on pseudoknot types.

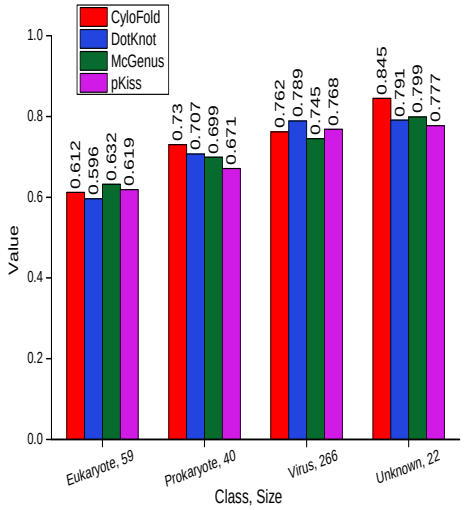
Figure 6.13: The sensitivity of the predictions by DotKnot, pKiss, CyloFold and McGenus.



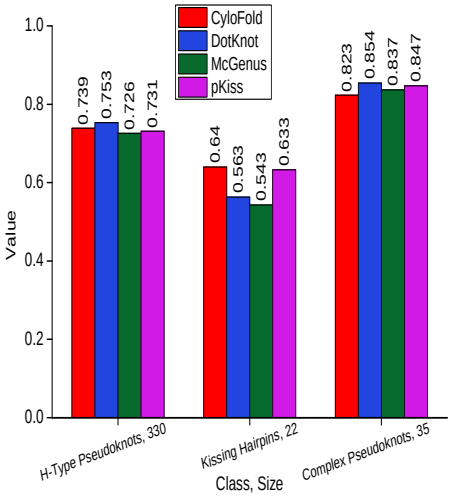
(a) Based on global groups.



(b) Based on the length and page no. of the sequences.

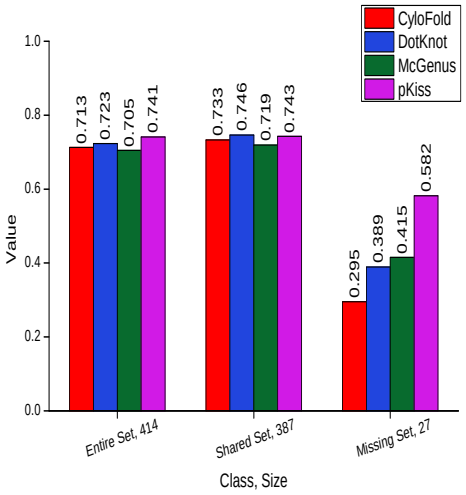


(c) Based on organisms.

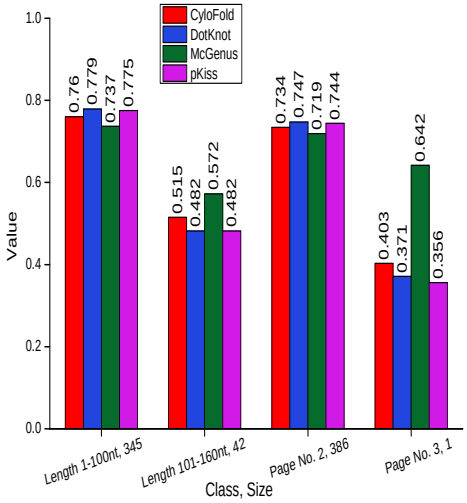


(d) Based on pseudoknot types.

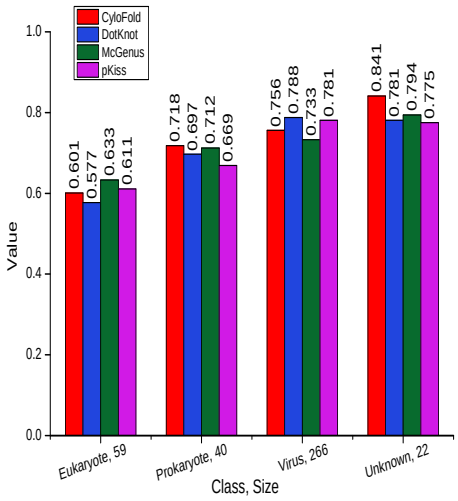
Figure 6.14: The PPV of the predictions by DotKnot, pKiss, CyloFold and McGenus.



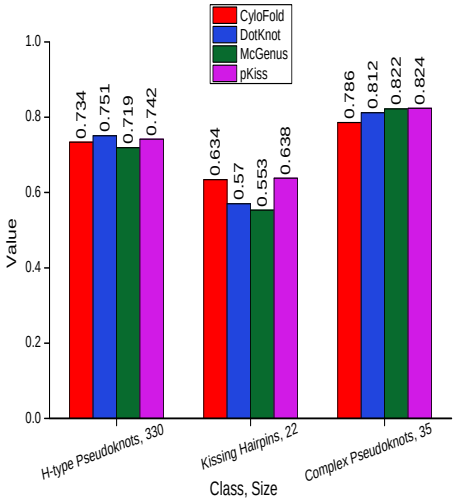
(a) Based on global groups.



(b) Based on the length and page no. of the sequences.



(c) Based on organisms.



(d) Based on pseudoknot types.

Figure 6.15: The MCC of the predictions by DotKnot, pKiss, CyloFold and McGenus.

In addition, Figure 6.16 shows the average performance on predicting each class of pseudoknots by the 15 benchmarking methods. Particularly, the classes are labeled in the x-axis, and the evaluation values ranging from 0 to 1 are labeled in the y-axis.

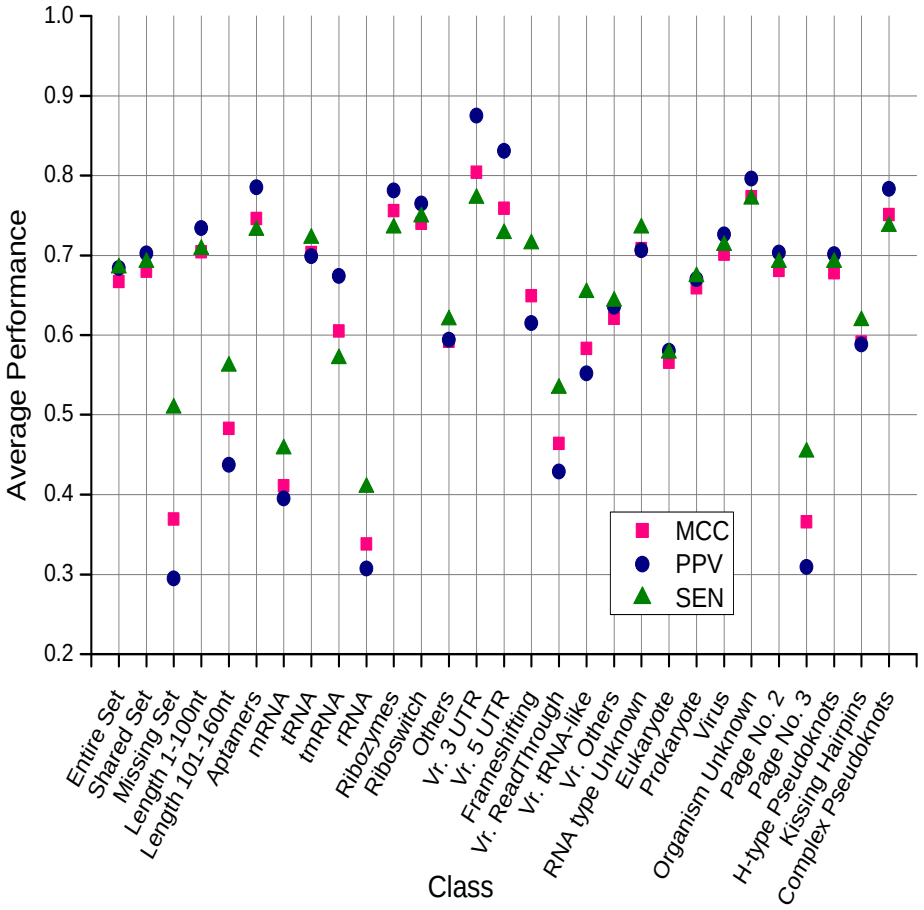


Figure 6.16: The average sensitivity, PPV and MCC upon the classes.

6.2.2 Individual Predictions

As Figures 6.5, 6.6 and 6.7 show the global performance of the predictions based on the 387 sequences in the shared set, we may still wonder how about the predictions based on the 27 sequences in the missing set.

Table 6.8 shows the detailed information of the 27 missing pseudoknots, including the length, pseudoknot type, page number, RNA type and organism, in the descending order of their lengths.

Figures 6.17, 6.18 and 6.19 tell the individual sensitivity, PPV and MCC details of predicting the 27 missing sequences by the 15 methods. Particularly, the failures of the prediction by each method are marked in yellow, namely the method fails to return a secondary structure for the given sequence.

Table 6.8: The 27 *missing* sequences.

Name	Length	Pseudoknot Type	Page No	RNA Type	Organism
3JYX_5	3170	recursive	2	rRNA	Eukaryote
3KIY_A	2848	complex	4		Prokaryote
2WDL_A	2807	complex	4	rRNA	Prokaryote
3J20_2	1495	complex	3	rRNA	Prokaryote
3ZEX_B	1465	complex	3	rRNA	Eukaryote
PKB192	1248	simple H-type	2	Viral others	Virus
3J2C_N	927	recursive	2	rRNA	Prokaryote
PKB64	920	simple H-type	2	rRNA	Prokaryote
PKB239	412	simple H-type	2	Viral others	Virus
3IZ4_A	377	recursive	2	tmRNA	Prokaryote
PKB149	351	simple H-type	2	Ribozymes	Prokaryote
3IYQ_A	349	recursive	2	tmRNA	Prokaryote
PKB193	341	simple H-type	2	mRNA	Prokaryote
PKB129	313	simple H-type	2	rRNA	Prokaryote
PKB208	237	simple H-type	2	Viral 5 UTR	Virus
PKB209	234	simple H-type	2	Viral 5 UTR	Virus
PKB171	224	kissing hairpin	2	Frameshifting	Virus
PKB77	219	simple H-type	2	Ribozymes	Eukaryote
PKB150	212	kissing hairpin	2	Ribozymes	Prokaryote
PKB181	207	simple H-type	2	Viral 5 UTR	Virus
PKB354	190	complex	2	Ribozymes	Eukaryote
PKB358	190	complex	2	Ribozymes	Eukaryote
PKB324	181	complex	2	Ribozymes	Eukaryote
PKB323	180	complex	2	Ribozymes	Eukaryote
3ZEX_C	169	kissing hairpin	2	rRNA	Eukaryote
3PDR_A	161	simple H-type	2	Riboswitch	Prokaryote
PKB357	160	complex	2	Ribozymes	Eukaryote

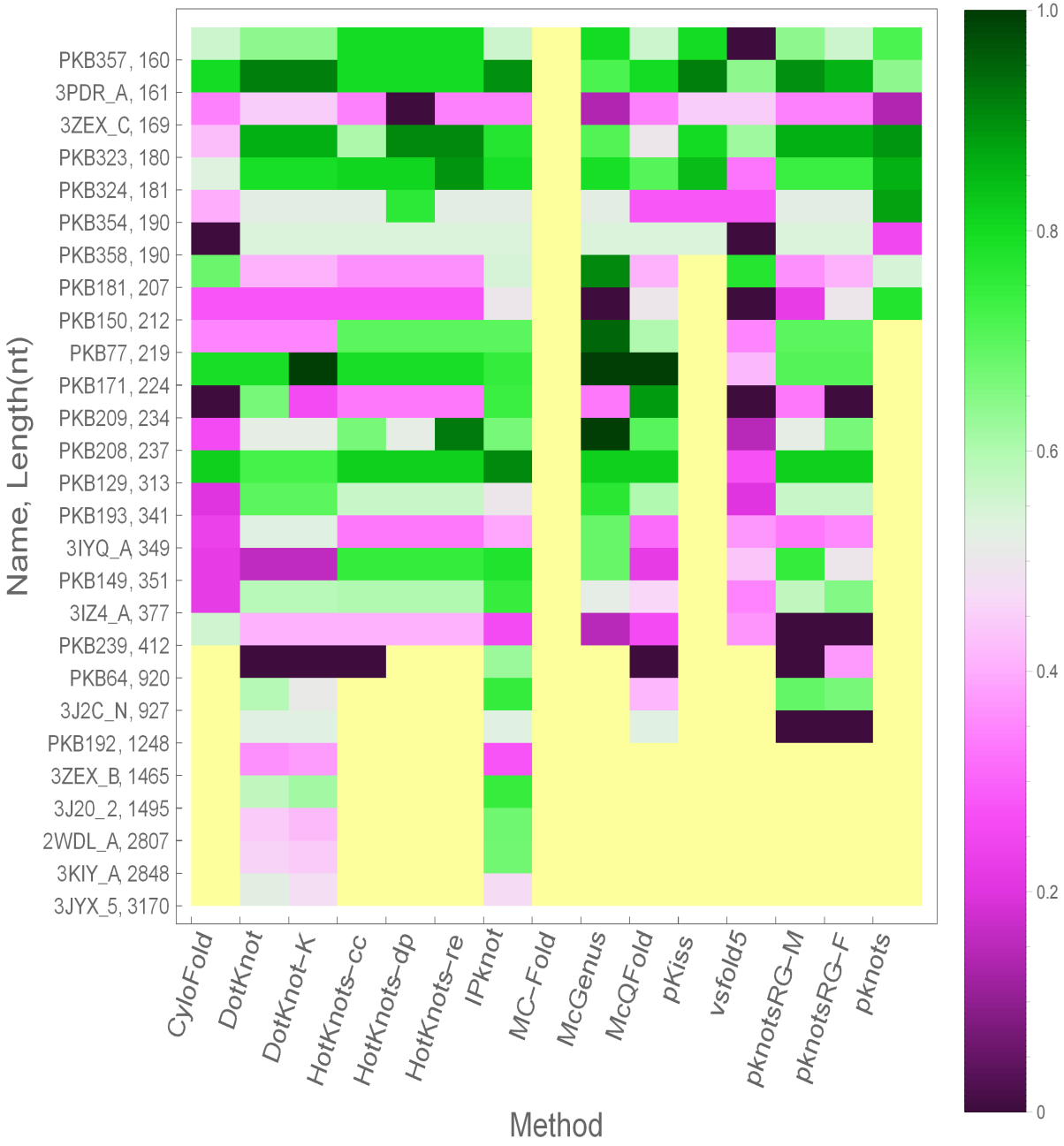


Figure 6.17: The density diagram of the sensitivity of the missing predictions.

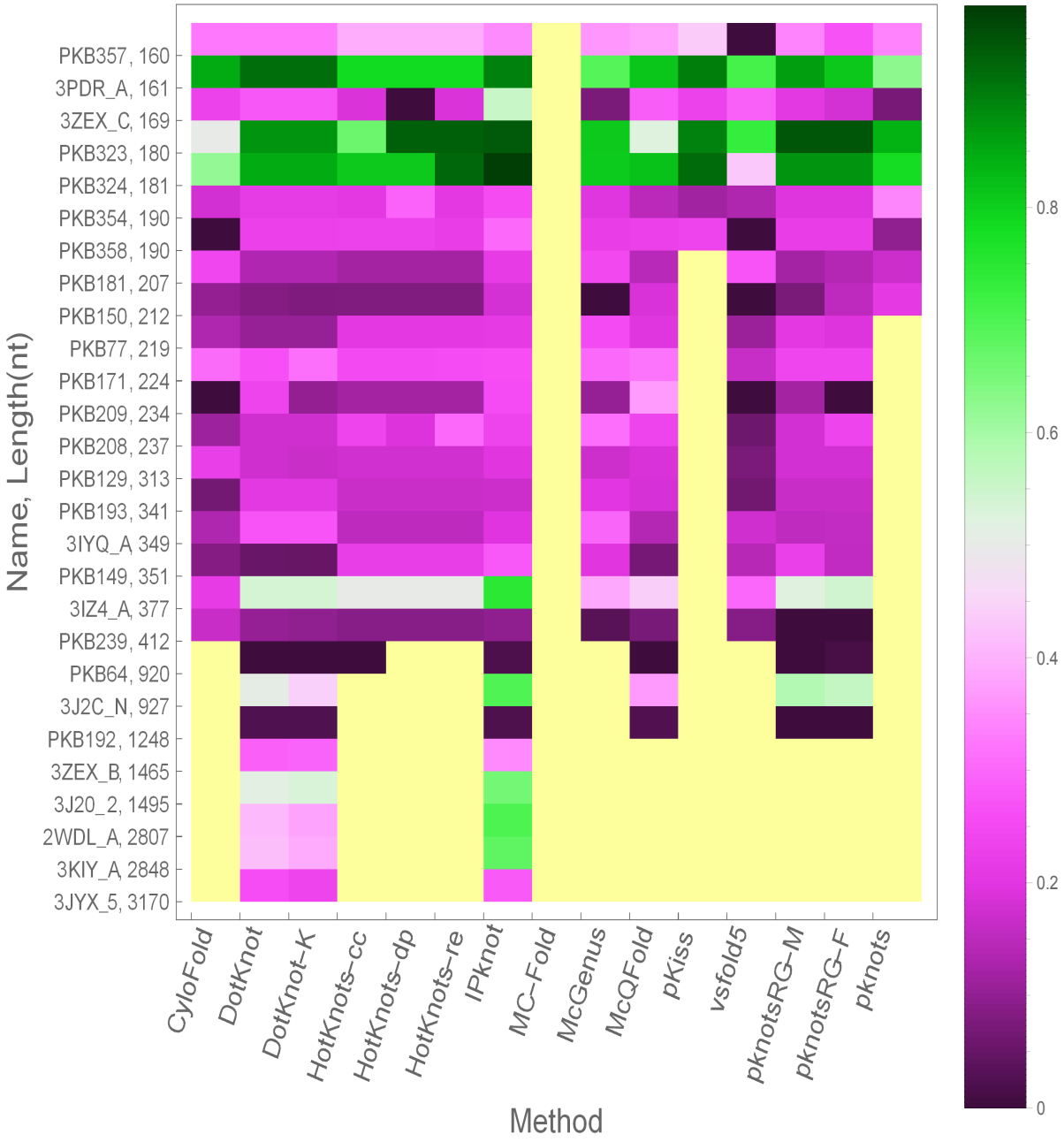


Figure 6.18: The density diagram of the PPV of the missing predictions.

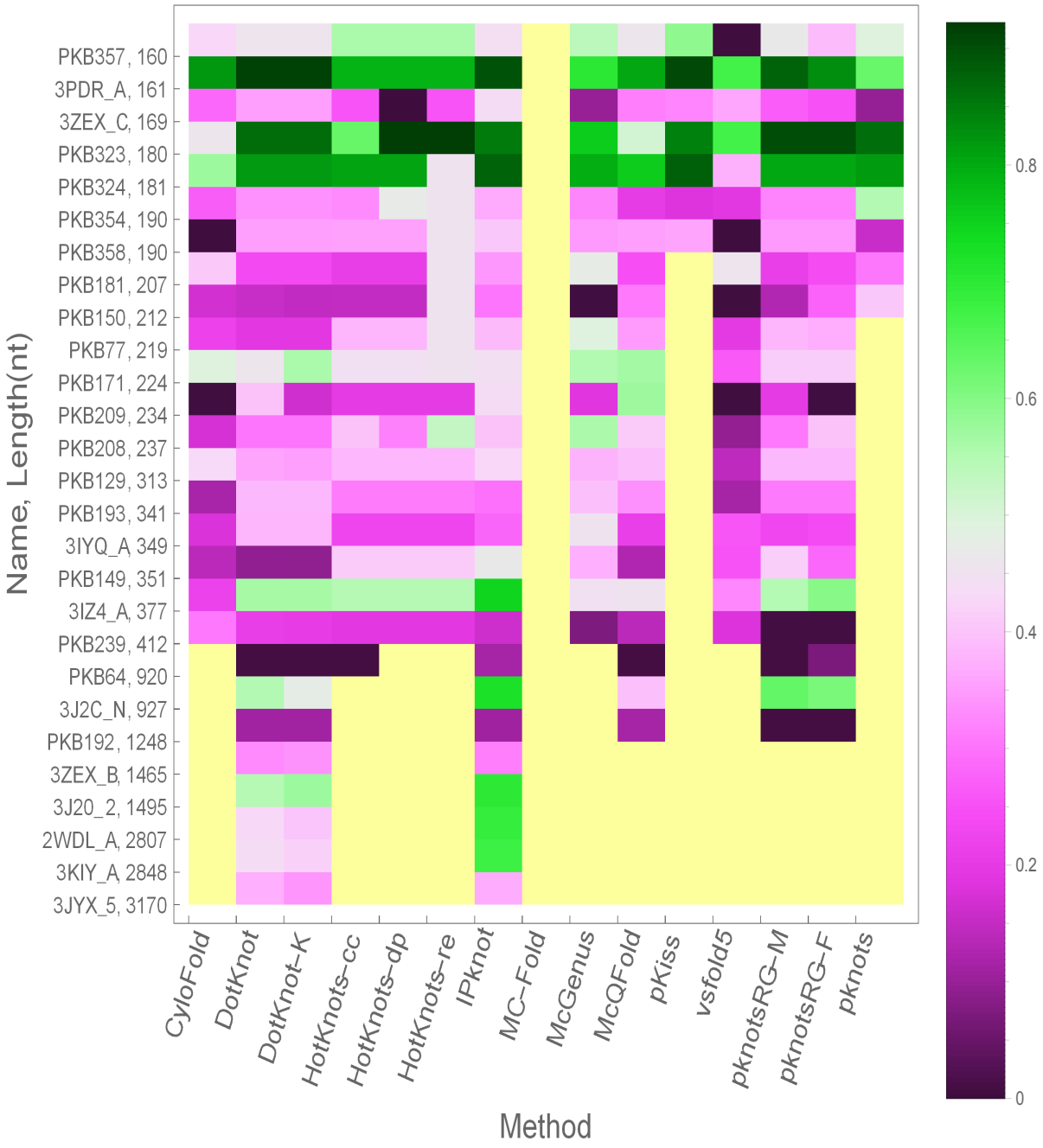


Figure 6.19: The density diagram of the MCC of the missing predictions.

The discussion and conclusion of the comparisons in this benchmark are going to be shown in the next chapter.

6.3 Web Development

This part of work is going to introduce the web development of the benchmark, providing the accessibility of the results of this dissertation to the researchers in the community.

Specifically, the web development of the benchmark includes three parts cardinally:

- What can be found in this on-line benchmark?
- How is the benchmark organized?
- How is the benchmark developed?

6.3.1 Functionalities

The main data that the benchmark has referred to are two datasets of pseudoknots, three complexity measures of pseudoknots, 15 secondary structure prediction methods, and three evaluation parameters. The results part includes the classification of pseudoknots, and the prediction of pseudoknots.

As a result, the on-line benchmark is going to consider the following sections, which are going to be represented in the main menu:

- Introduction
- Dataset
- Characteristics of pseudoknots
- Method
- Evaluation parameter
- Result
  - Classification of pseudoknots
  - Prediction of pseudoknots
- Manual

- Feedback

The *introduction* conveys a global description of this benchmark, covering the background of RNA secondary structure prediction, the motivation and main contributions of this benchmark. Quite remarkably, the benchmark first considers 414 pseudoknots from two prevalent databases, and analyses the hierarchical subsets of them based on three classifications, emphasizing on the page number particularly. Second, the on-line version of the benchmark supports the download and visualization of pseudoknots, and the querying of interested ones according to their length, RNA type, organism, pseudoknot pattern and any of the three pseudoknot classifications. Last but not least, 15 prediction methods which are available to predict pseudoknots are introduced, and their performance on predicting the 414 pseudoknots are compared, which is expected to help the users make a practical selection of prediction methods for the given sequence.

In the *dataset* part, the 414 pseudoknots from Pseudobase and PDB are listed. Specifically, the users can find the reference number of the pseudoknots, the links back to the two databases, the length of the pseudoknots, and the RNA type, organism and nucleotide composition details, as well as the corresponding reference secondary structure. In addition, a quick and an advanced search for the interested pseudoknots are provided, according to the length, RNA type, organism, pseudoknot pattern and complexity of the pseudoknots.

The *characteristics of pseudoknots* cover the three complexity measures of pseudoknots, which suggests the classifications of pseudoknots according to the physical interactions of base pairs, the theoretical treatability of certain algorithms, and the conformational page number.

The 15 methods considered in this benchmark are introduced in the *method* part.

And the calculations of the sensitivity, positive predictive value and Matthews correlation coefficient are listed in the *evaluation parameter* part.

The *result* part shows the classification of the 414 pseudoknots in the *classification of pseudoknots* part, and the prediction of pseudoknots by the 15 methods in the *prediction of pseudoknots* part.

The *manual* is a tutorial brochure, suggesting where the users can obtain the

information they are looking for, and a guide of the proper utilization of this benchmark.

And the *feedback* collects the comments and suggestions from the users, ensuring an improved and continuous support for this benchmark by us.

6.3.2 Architecture

Figure 6.20 shows the work-flow of this on-line benchmark. The 414 sequences in the datasets are classified into subsets according to three classifications of pseudoknots. Meanwhile, the 414 sequences are returned with one secondary structure by 15 prediction methods. Further, the predicted structures are compared with the references structures, assigning each prediction the evaluation values of true positives, true negatives, false positives, false negatives, sensitivity, PPV and MCC. We take the latter three as the prime evaluation parameters in this benchmark. The average sensitivity, PPV and MCC on the prediction of each class are calculated, voting a winner method which has obtained the optimal performance based on the particular class. Meanwhile, a consensus ranking based on all the classes is implemented, in order to vote the winner methods globally.

The corresponding entity relationship diagram of the data in the benchmark is shown in Figure 6.21. Specifically, the tables *True Negatives*, *False Negatives*, *True Positives*, *False Positives*, *PPV* and *MCC* have a similar table structure with the table *Sensitivity*. And the tables *Average_PPV* and *Average_MCC* have a consistent table structure with the table *Average_Sensitivity*.

6.3.3 Accessibility

In practice, the benchmark is built upon the framework of WordPress [Mullenweg et al., 2011], which is web software based on PHP and MySQL.

The web site is located on the server of LRI, and accessible to the bioinformatics community at: http://bernard-pk.lri.fr/, where the *BERNARD-PK* stands for *a BEnchmark for RNA stRucture preDiction with PseudoKnots*. And the screenshot of the home page is as follows:

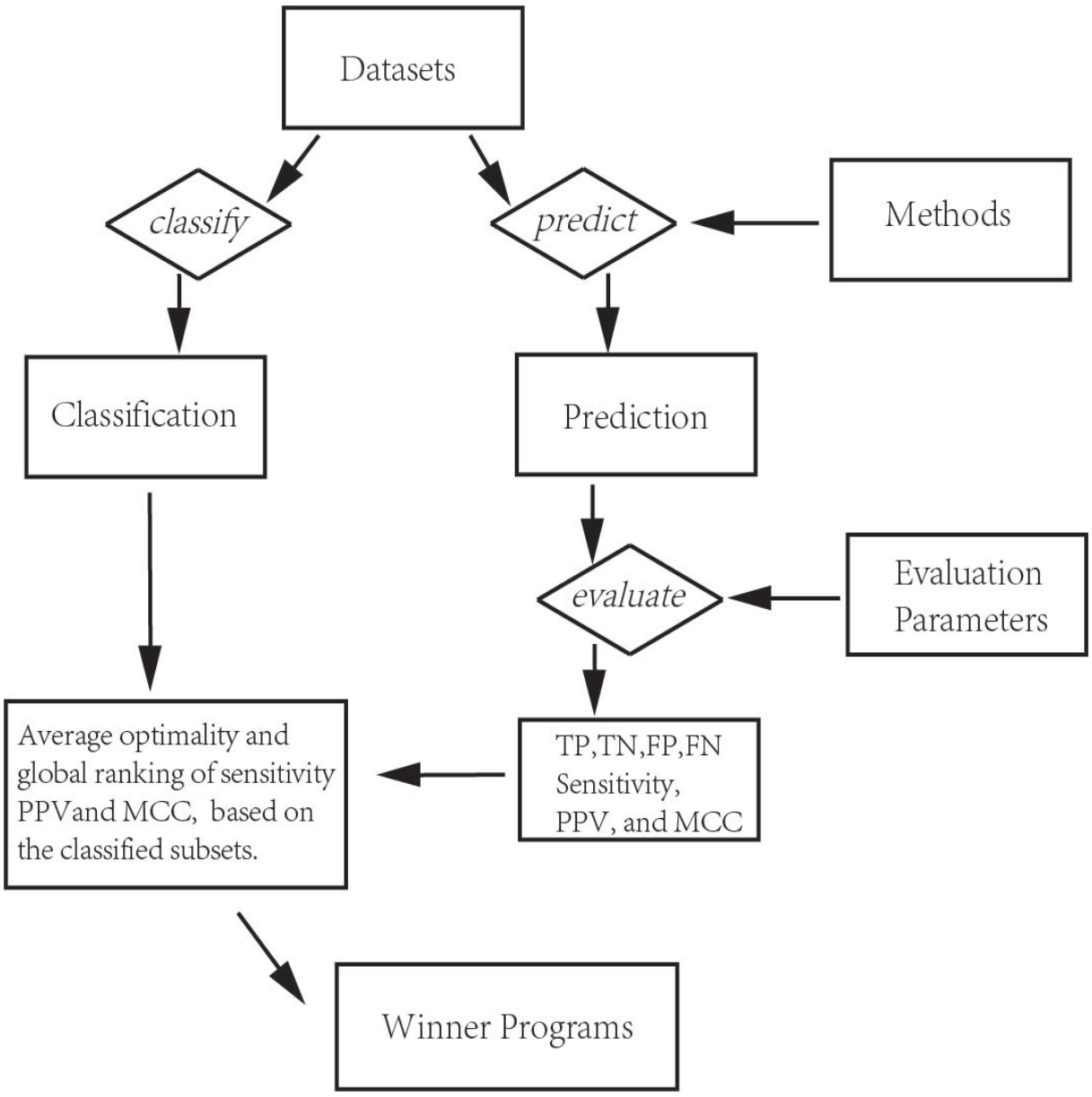


Figure 6.20: The work-flow of the benchmark.

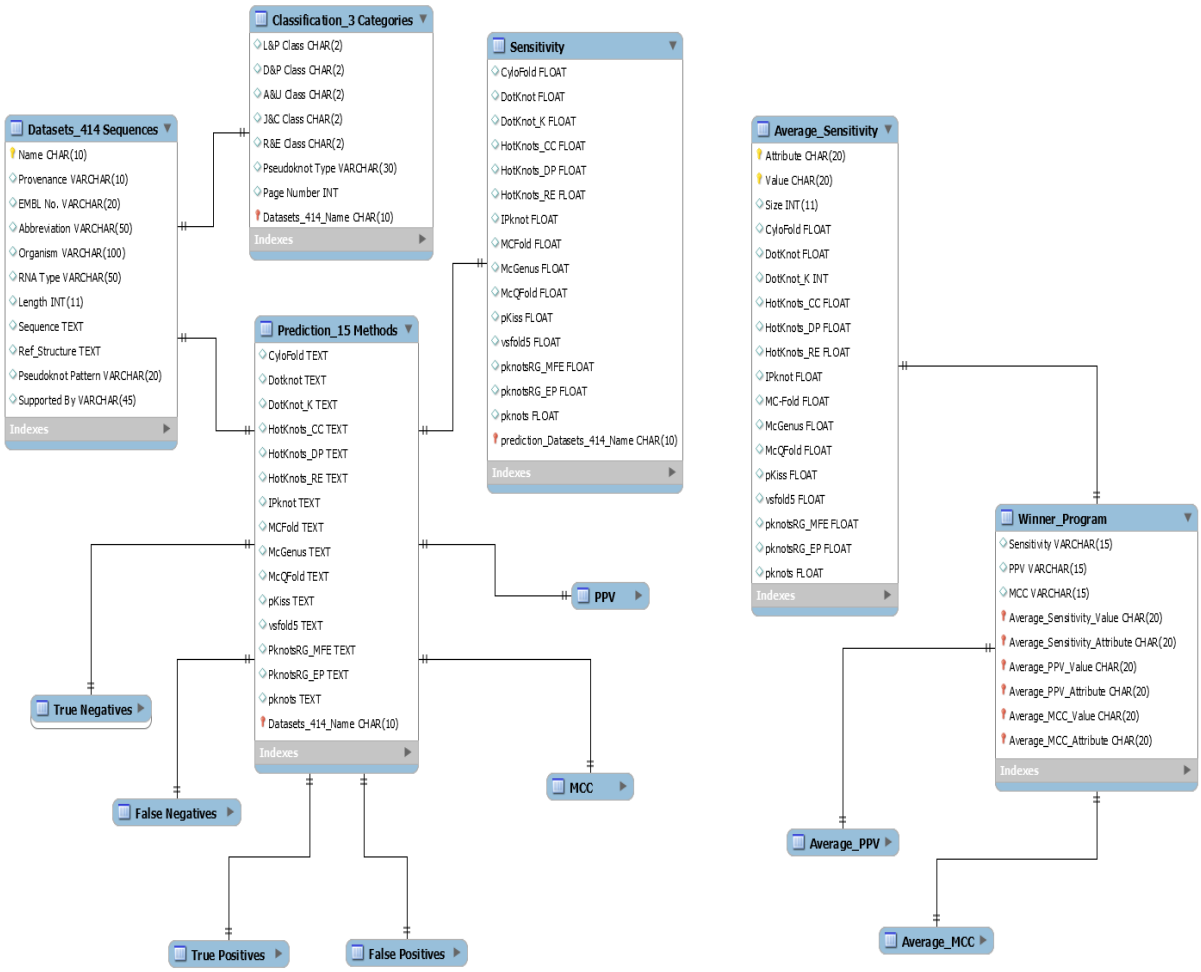


Figure 6.21: The entity relationship diagram of the tables in the benchmark.

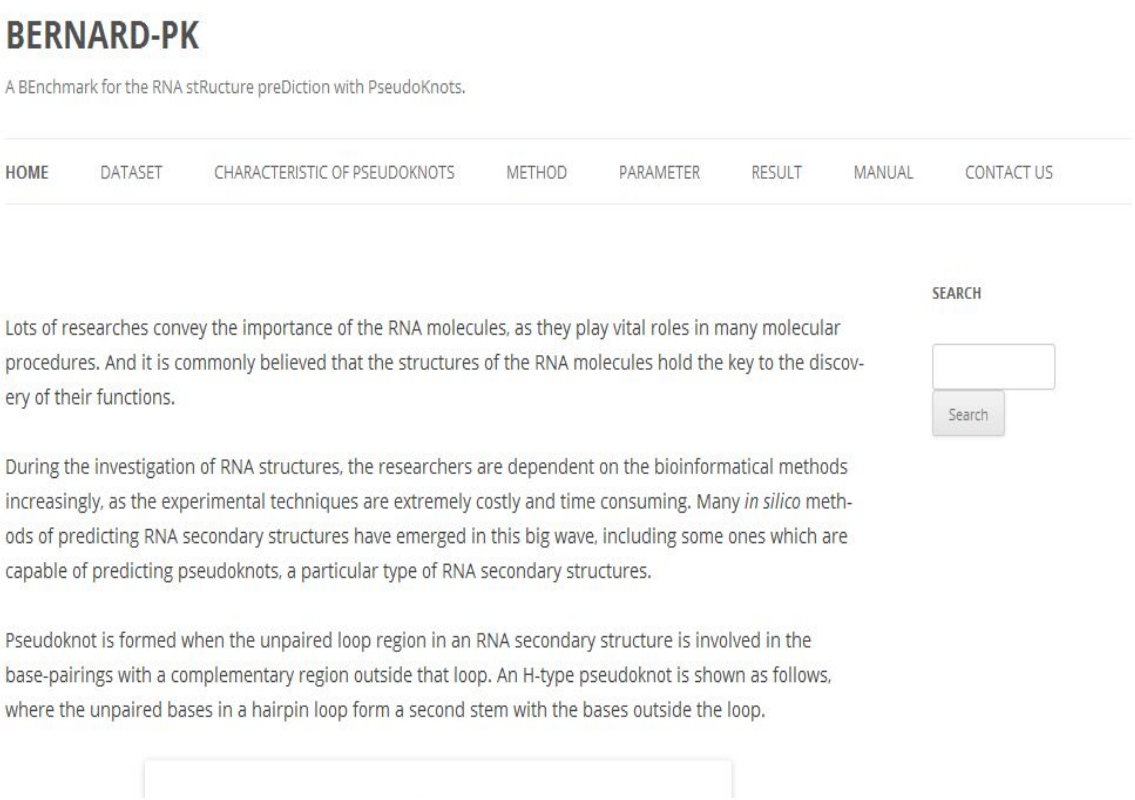


Figure 6.22: The on-line version of this benchmark.

Chapter 7

Discussion

7.1 Discussion

7.1.1 Pseudoknots Classification

As shown in Section 6.1, we may associate the relationship between the classifications of the *Physical Interactions* and the *Conformational Characteristics* in Table 6.1. The 409 pseudoknots having a page number of 2 include all the H-type pseudoknots, the kissing hairpins and recursive pseudoknots, and most part of the complex family. The excepted five complex pseudoknots with page number ≥ 3 contain more intricate crossing interactions.

This part of discussion pays more attention to the relationship between the physical classification and the algorithmic one, which is principally based on the discussion of the number of pseudoknots in the complementary set of each algorithmic class, as shown in Table 6.1.

It is very interesting to mention that the recursive pseudoknots defined in this benchmark are different from those defined in the algorithmic classification. This benchmark classifies the pseudoknots based on the RNA shadows, which removes all the non-crossing arcs from the original structure, collapses all unpaired bases, and replaces all adjacent parallel arcs by single arcs. The recursive pseudoknots in this context correspond to the set of pseudoknots which may include a second pseudoknot embedded in the unpaired single-strand region of them locally, as introduced in the Section 5.3.1. Contrarily, the algorithmic classification defines the

set of recursive pseudoknots as the ones containing some embedded substructures, which can either be the pseudoknots or the pseudoknot-free secondary structures. For example, given a pseudoknot with the pattern of *ABAcddcB*, the algorithmic classification will notate it as a recursive pseudoknot as there is a embedded substructure *cddc* embedded in the unpaired region of the pseudoknot *ABAB*. But the RNA shadow will notate its pseudoknot pattern as *ABAB* since the substructure *cddc* is nested, which is removed according to the definition of RNA shadows.

As a consequence, the RNA shadows are preferred to classify the pseudoknots physically as the 414 pseudoknots considered in this benchmark may display some intricate conformation.

H-type Pseudoknots and *L&P Class* of Pseudoknots

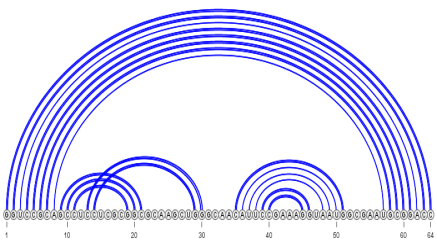
The conflict of defining recursive pseudoknots is specifically reflected on the different numbers between the H-type pseudoknots in the *Physical Interactions* and the *L&P class* of pseudoknots in the *Algorithmic Accessibilities* in Table 6.1, where the two numbers are supposed to be the same. More precisely, there are seven pseudoknots which belong to the H-type pseudoknots but not to the *L&P class*, the PKB65, 3JOL_A, 3NKB_B, 3PDR_A, 3SD1_A, 3U4M_B, 4FRG_B, and 4JRC_A.

Figure 7.1 shows four representative examples of the seven pseudoknots. It is clear that the four examples all contain some nested substructures inside or outside the H-type pseudoknots. We agree with the conclusion that the RNA shadow does lose the information on the size of the stems and non-crossing components of the global structure, although it captures the main crossing interactions of the pseudoknots.

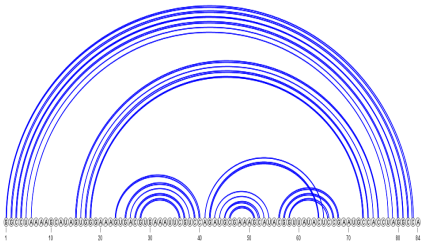
Typically, 3JOL_A and 3SD1_A shown in the Figures 7.1(c) and 7.1(d) do not have a consecutive stem to construct the pseudoknot, which may prevent them from falling into the *L&P class* further.

D&P Class of Pseudoknots

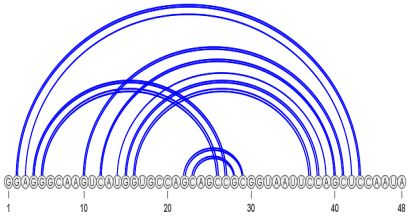
On the other hand, the *D&P class* of pseudoknots allows any number of H-type pseudoknots and their arbitrary concatenation and embedment inside each other.



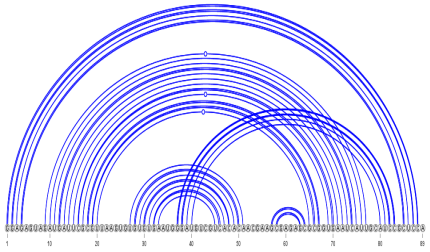
(a) The linear model of 3NKB_B



(b) The linear model of 4FRG_B



(c) The linear model of 3JOL_A



(d) The linear model of 3SD1_A

Figure 7.1: The H-type pseudoknots that do not belong to the *L&P class* of pseudoknots.

So the 344 *D&P class* of pseudoknots are composed of 341 H-type pseudoknots and three recursive H-type pseudoknots with their pseudoknot patterns shown in the Table 6.3.

A&U Class of Pseudoknots

As shown in Table 6.1, the number of pseudoknots in the relative complementary set of the *D&P class* in the *A&U class* is zero. It means that there is no pseudoknot in this benchmark which falls into the *A&U* class but does not belong to the *D&P class*.

We may wonder why? Is the *A&U class* of pseudoknots supposed to include a large number of simple pseudoknots and recursive simple pseudoknots which are composed of two stems? And what is the difference between the 35 complex pseudoknots shown in Table 6.4 with the pattern of *ABCDCADB* and the typical simple pseudoknot in the *A&U class*, which is shown in Figure 5.3(d) with the

pattern of *ABCBDADC*?

If we go back to the definition of the *A&U class* in Section 5.3.2, we have the explanation. In fact, the Akutsu's terminology of simple pseudoknots contains two crossing stems. The right bases of the first stem and the left bases of the second stem are interleaved arbitrarily, but the other bases all lie outside the interleaved area. Figure 5.3(c) shows the schematic diagram of the pseudoknot model in the *A&U class*. In other words, the pseudoknot model in the *A&U class* divides the sequence into three parts, the left region, the middle region, and the right region. The base pairs having a base on the left region can not have their partner on the right region, but on the middle region only, vice versa.

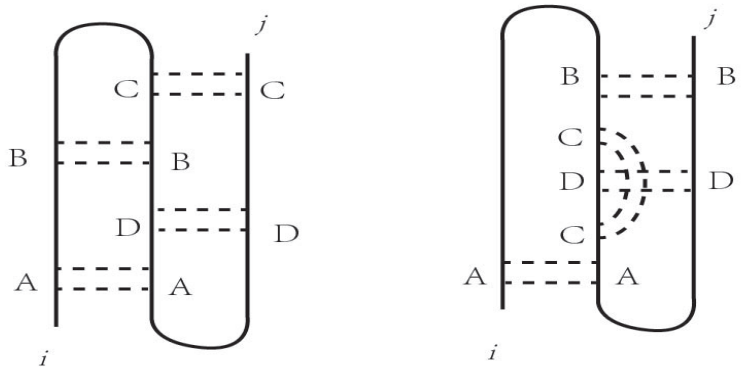
We try to decompose the 35 complex pseudoknots with the pattern of *ABCD-CADB* in Table 6.4 into the pseudoknot model of the *A&U class*. The base pair *BB* locates its 3' end at the end of the pseudoknot, which represents its location on the right region of the sequence. So the 5' end of *BB* should be on the middle region. On the other hand, the location of the 5' end of *BB* is close to the beginning of the pseudoknot, which suggests the embedded base pairs *CC* and *DD* both locate on the right part of the pseudoknot. But the overlap between the *CC* and *DD* makes it impossible to decompose the pseudoknot in the model of the *A&U class*, as shown in Figure 7.2(b), with respect to the rule that the base pairs on either left or right part of the *A&U class* of pseudoknots should not cross each other.

J&C Class of Pseudoknots

Next, the *J&C class* of pseudoknots corresponds to the density-2 pseudoknots, such as the H-type pseudoknots and the kissing hairpins. Consequently, the number of relative complementary set of the *D&P class* in the *J&C class* is 26, composed of 25 kissing hairpins and one recursive kissing hairpin, as shown in Table 6.3.

R&E Class of Pseudoknots

And there are 41 pseudoknots which fall into the *R&E class*, but neither into any of the previous classes. Specially, they are composed of all the complex pseu-



(a) The pseudoknots with the pattern of *ABCBDADC*.

(b) The pseudoknots with the pattern of *ABCDCADB*.

Figure 7.2: Comparison between the simple pseudoknots of the *A&U class* and the complex pseudoknots in Table 6.4.

doknots, except the 3ZEX_B, 2WDL_A and 3KIY_A, which do not belong to any algorithmic classes. The explanation of the exclusion of 3ZEX_B, 2WDL_A and 3KIY_A is unclear so far, as there is no precise description of the structure space of the textslR&E class of pseudoknots [Rivas and Eddy, 1999].

7.1.2 Prediction of the Pseudoknots

RNA Sequence Classes

The Section 6.2 shows the comparison of predicting the 414 pseudoknots by the 15 methods, based on the evaluation values which are assessed on the classes ranging from the entire set of 414 pseudoknots to each subclass inside the shared set.

As shown in Table 6.5, almost every method can handle the pseudoknots which are shorter than 200 nucleotides, with only one exception of MC-Fold as its threshold of length is shorter than 160 nucleotides. And the failure rate in handling the sequences by most methods increases as the length of sequence increases. For long sequences with more than 1000 nucleotides, most methods are incapable to predict them a secondary structure.

Similarly, the failure rate in handling the sequences by most methods increases as the complexity of pseudoknots grows. It is further supported by the fact that

nine complex pseudoknots and all the four recursive ones are excluded from the shared set. And two pseudoknots with a page number of 3 and both pseudoknots with a page number of 4 are excluded from the shared set. The only pseudoknot with a page number of 3 which is included in the shared set is the pseudotrefoil, PKB71.

Next, we are going to discuss the results in accordance of the classifications as shown in Table 6.2, based on Figures 6.5, 6.6 and 6.7.

In the classification of the global groups, as the entire set, shared set and missing set, the 15 methods have a close performance on predicting the entire set and the shared set, which is much higher than that of the missing set.

In the classification of the sequence lengths, we can see the performance based on the sequences with the length shorter than 100 nucleotides is better than that of the longer ones. Particularly, the performance gap between these two classes is much larger in the PPV and MCC values, compared to the sensitivity, suggesting the conclusion that the PPV and MCC values are more sensitive to some classes than the sensitivity.

In the classification of the functional families that the sequences belong to, all the 15 methods have obtained the worst performance based on the mRNAs and rRNAs, and viral readthrough. And they have obtained the best performance based on the viral 3 UTR, viral 5 UTR, riboswitch, aptamer and ribozyme. Particularly, the PPV and MCC values based on the mRNAs and rRNAs are greatly lower, compared to the sensitivity, which correspond to the rows in a darker fuchsia in Figures 6.6 and 6.7.

In the classification of their organisms, we can see that the 15 methods have the best performance based on the sequences with their organisms unknown, and have a relatively bad performance based on viral RNAs, and then the one based on the prokaryotic molecules. All the methods have the worst performance based on the eukaryotic molecules.

In the classification of the page number of each sequence, the performance of the 15 methods based on the sequences with a page number of 2 is much better than that of the particular sequence with a page number of 3. This supports the statement mentioned above, the performance decreases as the complexity of

pseudoknots grows. Particularly, the performance gap between these two classes is also much larger in the PPV and MCC values, compared to the sensitivity.

And in the classification of the pseudoknot types, we can see the performance of the 15 methods decrease in the order of predicting the complex pseudoknots, H-type pseudoknots, and then the kissing hairpins.

The average performance of the 15 methods can be referred to Figure 6.16. Typically, we can obtain the correlation between the evaluation parameters in Figure 6.16, as the three evaluation points gather in most classes.

On the other hand, Figures 6.5, 6.6 and 6.7 have an agreement that HotKnots-dp, HotKnots-re and IPknot have a common difficulty in predicting the viral readthrough sequences, and vsfold5 is a globally poor method in predicting all the classes. And Figures 6.6 and 6.7 vote MC-Fold as another poor method, as it has obtained a relative good performance of sensitivity. Contrarily, CyloFold, DotKnot, DotKnot-K, McGenus, pKiss and pknots are the relatively good prediction methods.

Further, as elaborated in Section 6.2.1, DotKnot, pKiss, CyloFold and McGenus are selected as four winner methods of this benchmark. Figures 6.10, 6.11 and 6.12 show their performance of predicting the functional families of the 387 sequences in the shared set. According to these three figures, we can see the four best programs perform well on predicting the sequences with their functional families of the aptamer, tRNA, riboswitch, ribozyme, viral 3 UTR, viral 5 UTR, and frameshifting. And the four programs have a relatively bad performance based on the prediction of the mRNA and rRNA. Typically, their performance always do not agree on the prediction of viral readthrough sequences, with respect to the sensitivity, PPV and MCC values.

Figures 6.13, 6.14 and 6.15 show the performance of DotKnot, pKiss, CyloFold and McGenus on predicting the other classifications of sequences, according to Table 6.2.

In the classification of the global groups, the four methods have obtain a comparative performance on predicting the entire set and the shared set, where a comparative performance suggests the closeness of evaluation values obtained by the four methods. While their performance varies much larger based on the miss-

ing set, but pKiss is always the best one to predict the 27 missing sequences, compared to the other three.

In the classification of the sequence lengths, the four methods have a generally better and comparative performance on predicting the sequences with the length shorter than 100 nucleotides, compared to the ones with a longer length. In the classification of the page number of each sequence, the four methods have a generally better and comparative performance on predicting the sequences with a page number of 2, compared to the sequence with a page number of 3. And both classifications vote McGenus as the best program of predicting the longer sequences and the pseudoknot with a page number of 3.

In the classification of their organisms, the performance of the four winner programs decreases slightly in the order of predicting the sequences with their organisms unknown, viral RNAs, prokaryotic molecules, and then the eukaryotic molecules. This tendency of losing advantages on prediction is consistent with that of the 15 methods based on the same classification of sequences.

And in the classification of pseudoknot types, the performance of the four winner programs decrease in the order of predicting the complex pseudoknots, H-type pseudoknots, and then the kissing hairpins, which is also consistent with that of the 15 methods based on the same classification of sequences.

If we move to the Figures 6.17, 6.18 and 6.19 for the individual prediction based on the 27 missing sequences outside the shared set, we may notice that the evaluation values for predicting the pseudoknots from PseudoBase are generally lower than the ones for predicting the pseudoknots from PDB.

This phenomenon can be explained by the incomplete information provided in the PseudoBase partially. Out of the 367 sequences in PseudoBase, there are 27 ones with different levels of structural information omitted, with an example PKB171 shown in Table 4.6, where the ':::' represents the unknown potential details in PseudoBase, and Figure 4.10. And the reason is that PseudoBase focuses on the crossing interactions forming the pseudoknots, and omits the partial structural details elsewhere. Consequently, in such cases, the inconclusive potential local substructures are referred to as unpaired bases in this benchmark, which makes the reference structures a conformation of a local pseudoknot and a long

unpaired free loop potentially. This operation decreases the total number of base pairs in the reference structures. As the consequence, the sensitivity of a prediction can be increased as the false negatives may decrease, and the PPV and MCC values can be decreased as the false positives may increase.

The evaluation values for predicting PKB64 may illustrate this phenomenon. The reference structure of PKB64 constructed in this benchmark contains only 8 base pairs, with 920 nucleotides. This makes the PKB64 very difficult to be predicted an acceptable secondary structure by the benchmarking methods, as extremely few information of the conformation is shown. The density of the corresponding boxes in Figures 6.17, 6.18 and 6.19 is in the darkest fuchsia. In other words, several methods can handle PKB64, but probably predict it a very different secondary structure. We can not tell how about the predictions as we have far from enough details to count the correct number of true positives and false negatives etc. of the prediction.

Prediction Methods

Table 6.6 concludes the winner program of the Figures 6.5, 6.6 and 6.7, with the highest evaluation values on average upon different classes, which are based on the 387 sequences in the shared set.

Globally speaking, pKiss is the one of best programs of this benchmark, as it obtains the highest sensitivity, PPV and MCC values for the 414 sequences in the entire set, although it has a restriction on the length of the input sequence which should be shorter than 207 nucleotides. It raises the possibility that pKiss may handle less number of pseudoknots, but it may return a secondary structure to the given sequence which is close to the reference structure once it is available.

More precisely, pKiss has obtained the optimal sensitivity, PPV and MCC for some certain classes, such as the ribozymes, viral 5' and 3' UTR under the classification of the functional families of the sequences. Meanwhile, McGenus, DotKnot have obtained the optimal sensitivity, PPV and MCC values for some certain classes as well.

However, for the rRNA, tRNA, viral molecules, there is no single method which has obtained the optimal sensitivity, PPV and MCC. In fact, according to

Equation 5.3, MCC combines the sensitivity and PPV, suggesting it as the most comprehensive evaluation parameter of the three ones.

As a consequence, we consider the method which has an optimal MCC and either an optimal sensitivity or PPV as the optimal method in predicting the current class. But there are two extremely special cases, the optimal sensitivity, PPV and MCC for the prokaryotic molecules and complex pseudoknots are obtained by three inconsistent prediction methods, as shown in Table 6.6.

In the purpose of investigating the prediction of the two special classes by the corresponding three winner methods, Table 7.1 shows the detailed evaluation scores that each method has obtained respectively, based on the particular classes of sequences.

Table 7.1: The performance of predicting prokaryotic molecules and complex pseudoknots by certain methods.

The Prediction for the Prokaryotic Molecules								
Sensitivity			PPV			MCC		
McGenus	McQFold	CyloFold	McGenus	McQFold	CyloFold	McGenus	McQFold	CyloFold
0.75	0.694	0.727	0.669	0.738	0.73	0.712	0.703	0.718
The Prediction for the Complex Pseudoknots								
Sensitivity			PPV			MCC		
McGenus	DotKnot	pKiss	McGenus	DotKnot	pKiss	McGenus	DotKnot	pKiss
0.821	0.783	0.811	0.837	0.854	0.847	0.822	0.812	0.824

The performance of predicting the two special classes by the corresponding three winner methods is comparable, from which we do not see any great difference of the evaluation values. We may suggest that McGenus, McQFold and CyloFold are all good choices in predicting the prokaryotic molecules with respect to the classification of the organism of the sequences. And McGenus, DotKnot and pKiss are good choices in predicting the complex pseudoknots with respect to the classification of the pseudoknot types alternatively.

Further, Table 6.6 and Figure 6.8 select three specific winner programs, DotKnot, McGenus and pKiss. And Figure 6.9 and Table 6.7 of the consensus ranking choose four global winner programs, CyloFold, DotKnot, McGenus and pKiss. We are wondering why CyloFold is a global winner program as its win counts in Figure

6.8 is four, which is far away from the other three winner programs?

In the investigation of the answer, we compare the rankings of these four winner methods based on 29 individual classes of Figure 6.7, with respect to the MCC values as an example. And typically, we focus on the comparison of rankings between CyloFold and McGenus principally, as their win counts of MCC in Figure 6.8 are two and eight respectively.

In fact, we notice that CyloFold has obtained 17 times of better MCC values than McGenus, while McGenus has 12 times of better MCC values than CyloFold. These superior MCC values obtained by CyloFold contribute it a cracking position in the consensus ranking, although CyloFold has obtained only two optimal MCC values in the 29 individual rankings.

This supports our conjecture that, there may be some method which performs globally great and stable but is hidden in the assessment of the optimal predictions. In fact, pknots is another underestimated method which has a good global performance as shown in Figure 6.9, but is not detective according to the Figure 6.8. Contrarily, McGenus is effective to a majority of classes, but may have an unsatisfactory performance in predicting other classes.

We may conclude that the consensus ranking assesses the performance of each prediction method more efficiently and comprehensively, as it recommends the winner programs based on a global assessment of the predictions. However, the specific winner methods selected from certain classes are significant as well, as they are more sensitive to the particular classes. The optimality of prediction may lose if we employ the global winner program to predict the certain class, rather than selecting the corresponding winner program as shown in Table 6.6.

In fact, the specific winner programs and the global winner programs are two pillars of the predictions in this benchmark, although they both contain DotKnot, McGenus and pKiss. They answer the motivation of this benchmark from two aspects, which of the prediction methods may effectively handle as many classes as possible, and which of the prediction methods may return an relatively best secondary structure for the given sequence?

A consequent recommendation of selecting a practical prediction method for the given sequence is as follows: Once certain information of the given sequence

is provided, the specific winner programs are recommended, or other particular method corresponding to the known information, according to Table 6.6. If there is no details about the given sequence, the global winner programs of CyloFold, DotKnot, McGenus and pKiss are strongly recommended.

In addition, MC-Fold and vsfold5, HotKnots-dp are the relatively poor prediction methods, based on the dataset of this benchmark.

Regarding the prediction of 27 sequences in the missing set, Figures 6.17, 6.18 and 6.19 show the performance of the individual predictions.

We may notice that IPknot displays its advantage in predicting the 27 sequences as it corresponds to the majority of green boxes in Figures 6.17, 6.18 and 6.19.

As the length of the sequence decreases, McGenus shows a good performance alternatively on some sequences. DotKnot, McQFold and pKiss can serve as an option as well. But after all, IPknot wins in this round of comparisons.

Additionally, the heuristic methods outperform the exact methods in almost all the comparisons, with the exception of vsfold5. This supports the main statement that the heuristic methods have a less restriction on the input length and pseudoknot type than the exact methods, but may bear a sacrifice on the optimality of the prediction. But the result shown in Table 6.6 and Table 6.7 oppose this declaration, arguing that the heuristic methods may predict a better conformation than the exact methods, and be more sensitive to the input sequences.

The Accuracy of the Predictions

We find out that pKiss is one of the best programs in this benchmark, as it has obtained dominant times of optimal evaluation values in Figure 6.8. Particularly, we investigate the predictions of pKiss in this part to start a discussion about the accuracy of the predicted structures. We try to answer the question: is pKiss always a reliable program?

As introduced in Section 5.4.2, pKiss is a program developed for predicting the kissing hairpins principally. In fact, after re-checking all the predictions of pKiss, I find that the 394 predictions are composed of 265 H-type pseudoknots, 95 kissing hairpins and 34 pseudoknot-free structures.

This discovery may disappoint the supporters of pKiss. pKiss predicts 72 more kissing hairpins than the reference structures as shown in Table 6.5. And it fails to predict any complex pseudoknots.

So another question may be raised again, how does pKiss achieve such high evaluation values? It may be explained as follows. pKiss has a relatively large group of the correctly predicted base pairs, but the global conformation is quite different from the reference structure.

This raises the third question. So how to evaluate the prediction? I am afraid it should be conquered by new evaluation parameters. In fact, as introduced in Section 4.3, the three criteria of sensitivity, PPV and MCC do not take care of the crossing interactions between base pairs, which are the most significant characteristic of pseudoknots.

Table 7.2 shows two different predictions as the examples, which may have the same evaluation values, both containing two predicted base pairs. Obviously, the predicted base pairs in both predictions are true positives. But two base pairs in the reference structure are missed, suggesting both predictions have their false negatives equal to 2, and false positives equal to 0. Meanwhile, the two predictions may have the same number of true negatives, which are computed according to the base pairs in the reference structure. The consistent numbers of TP, FN, FP and TN assign both predictions equivalent evaluation values. But theoretically *Prediction 1* may be the better prediction as it detects an analogous conformation to the reference structure, while *Prediction 2* predicts a conformation without any crossing interactions. This phenomenon screams for a more comprehensive evaluation system which takes the overlap of base pairs present in the pseudoknotted conformations into account.

Table 7.2: An example showing the flaw of the evaluation of two predictions.

Reference	`...(([[..).)..]]`
Prediction 1	`...(..[....)..].`
Prediction 2	`...((....).)....`

7.2 Conclusion

The last three chapters introduce a benchmark which focuses on the pseudoknots and the single-sequence prediction methods.

Chapter 5 introduces the motivation of this benchmark, as well as the preparation work. Specifically, Section 5.2 introduces the datasets used in the benchmark, Section 5.3 shows three main complexity measurements to classify the pseudoknots of the two datasets. Section 5.4 introduces 3 three exact methods and 12 heuristic methods involved in this benchmark. And Section 5.5 gives three evaluation parameters that are employed to assess the predictions returned by the methods.

Chapter 6 shows the results obtained, including both the classification of pseudoknotss, shown in Section 6.1, and hierarchical comparisons of predicting pseudoknots by the benchmarking methods, shown in Section 6.2.

Based on two sets of results, the respective discussions are aroused in Section 7.1 of the current chapter, which highlights the practical considerations for selecting an RNA pseudoknot prediction program. As the benchmark is accessible with an on-line version to the community, suggesting some web development details in Section 6.3 of this current chapter as well.

In addition, the benefits and lessons on selecting a practical prediction method that we obtain from this benchmark are concluded as follows:

- CyloFold, DotKnot, McGenus and pKiss are four best methods in this benchmark.

- The user may choose the specific winner programs, or a particular program for the given sequence according to the Table 6.6, if certain information is provided. In the majority cases, there should be one method which corresponds to the optimality in prediction. But if the optimal methods of sensitivity, PPV and MCC do not agree, such as the case in predicting the prokaryotic molecules and complex pseudoknots, the respective winner methods may be the alternative options to attempt. And for the sequences which are longer than 160 nucleotides, IPknot may be the first choice to try.

- If there is no details about the given sequence, the global winner programs

are recommended which are selected based on a global assessment of all the classes considered in this benchmark.

- If the user is interested in the local pseudoknots, KnotSeeker, or pknotsRG-loc are recommended, which are excluded from this benchmark as their unavailability of a global conformation.

- Both the quality of the reference structures and the reliable evaluation system may influence the comparison of predictions inestimably.

However, the 'best' program is the user's decision that depends on the RNA studied, the questions asked, the available experimental data and resources, and the intended applications of the structure prediction [Schroeder, 2009].

The efforts that this benchmark has made is trying to provide the user the useful information on how to choose a practical prediction program based on the statistical analysis of the 414 pseudoknots in this benchmark. But we offer no guarantee on an absolutely prefect recommendation of prediction method without any exceptions, as our conclusions are supported only by the optimality of the predictions by the 15 methods and based on the 414 sequences in this benchmark.

And frankly speaking, we don't expect that the winner programs in Tables 6.6 and Table 6.7 do know, or are capable to capture the crossing interactions very well. But we lay our confidence on the hypothesis that the sequences belong to the same subclass may display some analogous secondary structures. This hypothesis is supported by the theory that the sequences having a similar function may hold an analogue on the structural similarity as well. And we expect that the specific winner programs upon certain classes may capture this kind of analogues on the structures further.

Chapter 8

Conclusion and Perspectives

8.1 Conclusion

This dissertation focuses on the identification of pseudoknots, a secondary structural motif of RNA. It includes the study of the hierarchical classifications of pseudoknots, and the comparison of performance of the prediction methods that are available to predict pseudoknots from a single given sequence.

An RNA secondary structure without pseudoknots corresponds to a collection of nested base pairs, while the RNA pseudoknots are formed by the overlap or cross of the based pairs. The non-nested base pairs in the pseudoknotted conformation make their prediction much more complicated than that of the nested ones.

On the other hand, predicting an RNA secondary structure from the given sequence may employ diverse mechanisms. Minimizing the free energy of the RNA folding is the most prevalent strategy to predict an RNA secondary structures, which is implemented by dynamic programming algorithms and some heuristic strategies. Besides the thermodynamic stability, the probability of base pairs is frequently considered in many pseudoknot detection models. However it has been proved that predicting an RNA secondary structure containing arbitrary pseudoknots is NP-hard.

Specifically, we have conducted two rounds of researches of pseudoknots. The first one is based on the pseudoknots involved in the programmed ribosomal frameshifting, one typical recoding event which occurs in the regulation of post transcription. And the second one is based on the pseudoknots which participate

in more general molecular activities.

In practice, Chapter 4 describes the work on detecting the -1 programmed ribosomal frameshifting (-1 PRF) signals, where the ribosome switches to an alternative open reading frame by shifting one nucleotide to the upstream direction. The pseudoknots, one of the two main elements of a frameshifting signal, play the role in stimulating a frameshifting.

Principally, Orphea, KnotInFrame and PRFdb (a corresponding database storing their strong -1 PRF candidates), three methods detecting the -1 frameshifting signals were introduced, and their performance on predicting -1 frameshifting signals on three genomes were compared. Next, as another significant part of the comparisons, the former two programs were compared with their detection of frameshifting signals based on 34 frameshifting signals in PseudoBase.

Specifically, the sensitivity, positive predictive value (PPV) and Matthew's correlation coefficient (MCC), three evaluation parameters were introduced. A discussion about the further division of false positives was carried out as the consideration of the *compatible* set of false positives by the pioneered researchers. We finally decided to employ the standard equations to calculate the PPV and MCC, which do not consider the further separations, as the compatible false positives do not take the cross of base pairs into account.

According to the evaluation criteria, it has been shown that Orphea achieves a globally better performance than KnotInFrame as the corresponding evaluation values are higher than that of KnotInFrame.

Chapters 5, 6 and 7 introduce a benchmark on the much more general pseudoknots and prediction methods. Our motivation of this work was to guide the users to select a practical RNA pseudoknots prediction method with respect to the given sequence. In practice, we considered 414 pseudoknots which are the entries of PseudoBase and Protein Data Bank (PDB), and 15 state-of-the-art methods that are available to predict RNA pseudoknots, including three exact methods and 12 heuristic ones. The predictions of the 414 pseudoknots by the 15 methods were assessed with the consistent evaluation parameters with Chapter 4, and based on individual sub-collections of pseudoknots which are divided by hierarchical measurements, such as the length of the sequences, the RNA family they

are.

In addition, a detailed anatomy of the complexity of pseudoknots was introduced. This refers to the classifications of the 414 pseudoknots, with respect to the physical interactions of the crossing base pairs, the algorithmic accessibilities where a particular class of pseudoknots is defined as the set of structures that can be returned by the corresponding prediction method theoretically, and the conformational characteristics such as the page number of each pseudoknot. It has been proven that the calculation of page number for arbitrary pseudoknots is NP-hard.

The results of the classification of RNA pseudoknots show that the pseudoknots in nature have a relatively low value of complexity, such as the maximal page number of the 414 pseudoknots is 4. On the other hand, the results of comparing the prediction of pseudoknots by the 15 methods were concluded as follows.

We voted three methods as the specific winner programs, which have obtained the optimality in predicting some particular sub-collections of pseudoknots. And we voted four methods as the global winner programs, which have obtained the optimality in predicting all the 414 pseudoknots. We recommend the 'beneficiaries' of this benchmark to choose the specific winner programs or some others which are optimal to the certain sub-collection of pseudoknots, as some detailed information of the given sequence are known. The global winner programs can be effective if there is no details about the given sequence.

8.2 Perspectives

As shown in Section 6.2.1, the main work of the benchmark is to compare the prediction performance of the 15 methods, based on the 29 classes of sequences shown in Table 6.6, which are divided in accordance of the hierarchical classifications of sequences. The optimal programs for individual classes are chosen respectively according to the predicted evaluation values. Our first perspective is the consideration of selecting the optimal prediction method based on several classes concurrently. For example, we have selected McGenus as the best program both in predicting mRNA with respect to the classification of the functional fami-

lies of the sequences, and in predicting the eukaryotic molecules with respect to the classification of the organisms of the sequences. We are wondering: may McGenus achieve an optimality on predicting the mRNAs which are found in the eukaryotic molecules particularly, as shown in Table 6.2 in Section 6.1.2. This consideration is expected for increasing the practicality of the recommended prediction methods, which are chosen according to their optimality in a set of classes.

Second, as mentioned in Section 7.1.2, there are dozens of pseudoknots of PseudoBase whose secondary structural information is omitted partially as the database focuses more on the crossing interaction forming the pseudoknots. The incomplete details suggest a huge obstacle to compare the predicted structure with the reference. The extreme cases are the PKB64 and PKB192, with 920 and 1248 nucleotides respectively, whose secondary structural details are almost unknown except the pseudoknotted base pairs. This bothers the comparison of prediction outside this pseudoknot, namely the predicted base pairs should be classified as false positives or not and the non-predicted ones should be classified as false negatives or not. This dilemma suggests a new comparison of predicting pseudoknots by the contemporary methods once new and more comprehensive datasets are available.

Third, as mentioned in Section 5.4, the considered methods in this benchmark are not exhaustive. We hold an expectancy of a continuous evaluation of the RNA pseudoknots prediction methods. It relies on two aspects of efforts. The first one is to carry out the evaluation of prediction methods based on the newly released datasets, and the second one is based on the emerging prediction methods. In practice, we expect to provide a platform for the developers of the new methods, who may expect to upload the performance of their methods on predicting the 414 pseudoknots in this benchmark, and compare with that of the 15 benchmarking prediction methods.

Fourth, we also want to extend our benchmark as an automated recommender system, which is able to return the users a prediction method and the corresponding predicted secondary structure, with respect to the sequence and the descriptive details provided.

Last but not least, we show our explanations on the unsatisfactory preciseness

of the employed evaluation parameters in Section 4.4.4 and the last part of Section 7.1.2, as they do not consider the crossing interaction in the pseudoknots particularly. We neither have a more comprehensive conception of a new evaluation system so far, nor a plausible modification of the definitions of the compatible false positives, which are tolerated as acceptable prediction by the previous work. But we hold a strong confidence on the significance of this part of future work.

Appendix A

The Comparison of Predicting the Strong Candidates of Orphea

Table A.1: The Comparison of Orphea's 6 Best Predictions

Sequence Name	**Program**	**Result**
54_Random_0.179	Slippery Site	GUUAAAU
	SubSequence	UGGAGGCAGACAAAAAUUGGAAGAUCAAGCCCAUCUGCCUUUCAGUUGCCAUAGUCCAAUUU
	Orphea	...(((((((...[[[[[[[[.....)))))))).....]]]]]]]]
	KnotInFrame	No suitable slippery sites have been detected.
	CyloFold	.(((((((((...[[[[[[[[.....)))))))))....]]]]]]]]
	IPknot-2	.(((((((((......(([[......))..]].)))))))))..((((........))))..
	IPknot-3	.(((((((((......(([[......))..]].)))))))))..((((........))))..
	pknotsRG-M	(((.((((((......(((......)))).....))))))....[[[[]))).....]]]]].
	pknotsRG-F	(((.((((((......(((......)))).....))))))....[[[[]))).....]]]]].
	DotKnot-P	.(((((((((...[[[[[[[[.....)))))))))....]]]]]]]]
	DotKnot-K	.(((((((((...[[[[[[[[.....)))))))))....]]]]]]]]
	MC-Fold	(((((..))))).((((((((((((...(.(((((((...))))))))))))))))))))
	Vsfold5	..(((((((((......(((......)))).....)))))))..(((((........))))).
3406_Human_0.1332	Slippery Site	GGGAAAA
	SubSequence	UACGUGGGCACAGCAGUGAAUCCUCACACCCUGGGCUUUGCCCAAGAAGUGCUCGUGAACCUGAUGAAAGUGUCGGGUGU
	Orphea	....((((((.(((...........[[[[[[...))).)))))).....]]]]]]
	KnotInFrame	...(((((...[[[[.......)))))....(((((...)))))).....]]]].....
	CyloFold	..(((((((((....((((....))))....((((...))))).....)))))))).(((((((......)))))))..
	IPknot-2	..(((((((((....((((....))))....((((...))))).....)))))))).(((((((....)))))))..

Continued On Next Page

Table A.1 – *Continued From Previous Page*

Gene Name	Program	Result
3406_Human_0.1332	IPknot-3	..(((((((((....((((....))))....((((...))))).....)))))))).(((((((....))))))))..
	pknotsRG-M	..(((((((((....((((....))))....((((...))))).....)))))))).(((((((......))))))))..
	pknotsRG-F	....((((((.....((((....))))....[[[[[..)))))).....]]]]]...(((((((......))))))))..
	DotKnot-P	..(((((((((....((((....))))....((((...))))).....)))))))).(((((((......))))))))..
	DotKnot-K	..(((((((((....((((....))))....((((...))))).....)))))))).(((((((......))))))))..
	MC-Fold	
	Vsfold5	.....((((((....((((....))))....(((.....))))....))))))....(((((((......))))))))..
57_Random_0.131	Slippery Site	GUUUUUU
	SubSequence	GGAGGUCAGGGGUGUCAUUCUUGGGGUACCCCCCCCAAAUAUUUGUCCGUAUACUAUCAUUAUGCUCAACAAGGCCGAG
	Orphea	........(((((((...[[[[[)))))))).....................................]]]]]
	KnotInFrame	No suitable slippery sites have been detected.
	CyloFold	...((((.((((((......[[[[[.))))))]]]]].........((((.......)))))).......))))...
	IPknot-2	...((((..[[[[[......(((((]]]]])))))).........(((.........))))).......))))...
	IPknot-3	...((((..[[[[[......(((((]]]]])))))).........(((.........))))).......))))...
	pknotsRG-M	........(((((((...[[[[[)))))))).........((((..(((((.......)))))...)))))..]]]]]
	pknotsRG-F	........(((((((...[[[[[)))))))).........((((..(((((.......)))))...)))))..]]]]]
	DotKnot-P	........(((((((...[[[[[)))))))).........((((..(((((.......)))))...)))))..]]]]]
	DotKnot-K	(((.....[[[[[....)))(((((..]]]]])))))....((((..(((((.......)))))...)))).......
	MC-Fold	(((((((.[[[[[)))))))(((((((...)))))))((((((((((((((...))))))))))))))]]]]].
	Vsfold5	...((((.(((......))).((((.....)))))).....(((..(((((.......)))))...))).))))...

Continued On Next Page

Table A.1 – *Continued From Previous Page*

Gene Name	Program	Result
4335_Human_0.0881	Slippery Site	GGGAAAC
	SubSequence	GAGGCAGGGGCUGGGGCCAUGAGCCAUCCCAAGCCCUGGGAAACAUAGGCUCA
	Orphea	.......(((((.(((...[[[[[[..))).)))))..........]]]]]]
	KnotInFrame	........((((.[[[[....))))).......]]]].................
	CyloFold	..(((.[[[[[[...))).((((((.((((.]]]]]]]]))))......))))))
	IPknot-2	......(((((((((.............))).))))))..............
	IPknot-3	......(((((((((.............))).))))))..............
	pknotsRG-M	..(((...(((....)))....))).(((((.[[[.)))))......]]]...
	pknotsRG-F	..(((...(((....)))....))).(((((.[[[.)))))......]]]...
	DotKnot-P	..(((...[[[[...)))...]]]].(((.....[[[[)))....]]]]....
	DotKnot-K	..(((.[[[[[[...))).........((((.]]]]]]])))..........
	MC-Fold	((((((((((((((((..(((...))))))))))))))(((...)))))))))).
	Vsfold5	........((((.........)))).(((((.[[[.)))))......]]]...
1679_Human_0.0592	Slippery Site	GGGAAAC
	SubSequence	UCCCCGGCCCCGCUGUAGAGGGACCUUCAGCGACCGGGCCAGAAAUAAUAAGGUCCC
	Orphea	.....(((((.........[[[[[[[[........)))))).........]]]]]]]]
	KnotInFrame	No suitable slippery sites have been detected.
	CyloFold	.....((((([[[[[....{{{{{{{{]]]]]...)))))).........}}}}}}}}
	IPknot-2	.....((((((((((.((......)).)))))...))))).................
	IPknot-3	.....((((((((((.((......)).)))))...))))).................

Continued On Next Page

Table A.1 – *Continued From Previous Page*

Gene Name	Program	Result
1679_Human_0.0592	pknotsRG-M	.....((((((((((..(((....)))))))).)))))..............
	pknotsRG-F	.....(((((.((((..(((....)))))))[[[[)))))..........]]]]..
	DotKnot-P	.....(((((.........[[[[[[[[........)))))).........]]]]]]]]
	DotKnot-K	.....(((((.........[[[[[[[[........)))))).........]]]]]]]]
	MC-Fold	((((((((((((((((((.[[[[[[[[))))))))))))))))))....]]]]]]]]
	Vsfold5	..(((((...(((((....[[[[[[[[)))))).)))))..........]]]]]]]]
4339_Human_0.0558	Slippery Site	GGGAAAC
	SubSequence	GGGAACUGGGCUUGGGACAAGAGCCAUCCCAAGUCCAAGGCCAAGUAGGCUC
	Orphea	......((((((((((...[[[[[.))))))))))).........]]]]]
	KnotInFrame	........(((((.[[[[..)))))).......]]]]...............
	CyloFold	......((((((((((...[[[[[.))))))))))).........]]]]]
	IPknot-2	......((((((((((.........))))))))))..((((.....)))).
	IPknot-3	......((((((((((.........))))))))))..((((.....)))).
	pknotsRG-M	......((((((((((...[[[[[.))))))))))).........]]]]]
	pknotsRG-F	......((((((((((...[[[[[.))))))))))).........]]]]]
	DotKnot-P	......((((((((((...[[[[[.))))))))))).........]]]]]
	DotKnot-K	......((((((((((...[[[[[.))))))))))).........]]]]]
	MC-Fold	((((((((((((((((...[[[[[[))))))))))))))))...]]]]]]
	Vsfold5	......((((((((((.........))))))))))..((((.....)))).

Table A.2: The General Comparison of 49 predictions of Orphea.

Sequence Name	**KnotIn Frame**	**Cylo Fold**	**IPknot -N**	**IPknot -P**	**pknots RG-M**	**pknots RG-F**	**pknots RG-L**	**DotKnot -P**	**DotKnot -K**	**MCFold**	**Kine Fold**	**Prob Knot**	**Vsfold5**	**Scores (YES/13)**
Bracket[1]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	
54_Random _0.179	no	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	no	no	9/13
3406_Human _0.1332	yes	no	no	no	no	yes	yes	no	no		no	no	no	3/12
57_Random _0.131	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	10/13
4335_Human _0.0881	yes	yes	no	no	yes	yes	yes	yes	yes	no	yes	yes	yes	10/13
1679_Human _0.0592	no	yes	no	no	no	yes	yes	yes	yes	yes	yes	no	yes	8/13
4339_Human _0.0558	yes	yes	no	no	yes	yes	yes	yes	yes	yes	no	no	no	8/13
51_Yeast _0.0325	no	yes	no	no	no	yes	yes	no	no	no	no	no	yes	4/13

Continued On Next Page

Table A.2 – *Continued From Previous Page*

Sequence Name	KnotIn Frame	Cylo Fold	IPknot -N	IPknot -P	pknots RG-M	pknots RG-F	pknots RG-L	DotKnot -P	DotKnot -K	MCFold	Kine Fold	Prob Knot	Vsfold5	Scores
35_Random _0.0248	no	no	yes	yes	no	yes	yes	yes	yes	no	no	no	no	6/13
4_Yeast _0.0214(54)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	11/13
86_Yeast _0.0172	yes	yes	no	no	no	yes	yes	yes	yes	no	yes	no	yes	8/13
1455_Human _0.0156	no	yes	no	no	yes	yes	yes	yes	yes	yes	yes	no	no	8/13
6_Random _0.0155	no	yes	no	no	yes	yes	yes	yes	yes	yes	yes	no	yes	9/13
87_Random _0.0143	yes	yes	yes	yes	no	yes	yes	no	no	no	no	no	yes	7/13
27_Random _0.0133	yes	yes	no	no	no	yes	yes	yes	yes	yes	yes	no	no	8/13
48_Random _0.0127	no	no	no	no	no	yes	yes	no	no	no	no	no	no	2/13
226_Human _0.0105	no	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	10/13

Continued On Next Page

Table A.2 – *Continued From Previous Page*

Sequence Name	KnotIn Frame	Cylo Fold	IPknot -N	IPknot -P	pknots RG-M	pknots RG-F	pknots RG-L	DotKnot -P	DotKnot -K	MCFold	Kine Fold	Prob Knot	Vsfold5	Scores
19_Yeast _0.0102	no	yes	no	no	no	yes	yes	yes	yes	yes	no	no	no	6/13
263_Human _0.0096(54)	yes	no	no	no	no	yes	yes	yes	yes	no	no	no	no	5/13
4395_Human _0.009	no	no	no	no	yes	yes	yes	yes	yes	no	yes	no	yes	7/13
4287_Human _0.0088	no	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	no	yes	10/13
1539_Human _0.0088	yes	yes	no	no	yes	yes	yes	yes	yes	no	no	no	yes	8/13
3280_Human _0.0083	yes	yes	yes	yes	no	yes	yes	yes	yes	no	yes	no	yes	10/13
144_Yeast _0.0077	no	no	no	no	no	yes	yes	no	no	no	no	no	no	2/13
55_Yeast _0.0071	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	12/13
155_Yeast _0.0065	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	11/13

Continued On Next Page

Table A.2 – *Continued From Previous Page*

Sequence Name	KnotIn Frame	Cylo Fold	IPknot -N	IPknot -P	pknots RG-M	pknots RG-F	pknots RG-L	DotKnot -P	DotKnot -K	MCFold	Kine Fold	Prob Knot	Vsfold5	Scores
161_Yeast _0.0064	no	no	no	no	no	yes	yes	no	no	no	yes	no	no	3/13
2487_Human _0.0063	no	yes	yes	yes	no	yes	yes	yes	yes	no	yes	no	no	8/13
84_Random _0.0062	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	12/13
29_Yeast _0.0051	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	11/13
77_Random _0.0049	no	yes	yes	yes	no	yes	yes	no	no	yes	yes	yes	yes	9/13
2314_Human _0.0047	no	yes	yes	yes	no	yes	yes	yes	yes	no	yes	no	no	8/13
3_Random _0.0045	no	yes	no	no	no	yes	yes	yes	yes	yes	yes	no	yes	8/13
33_Random _0.0044	yes	yes	no	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	11/13
37_Random _0.0039(37)	no	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	no	yes	12/13

Continued On Next Page

Table A.2 – *Continued From Previous Page*

Sequence Name	KnotIn Frame	Cylo Fold	IPknot -N	IPknot -P	pknots RG-M	pknots RG-F	pknots RG-L	DotKnot -P	DotKnot -K	MCFold	Kine Fold	Prob Knot	Vsfold5	Scores
141_Yeast _0.0034	no	yes	no	no	yes	yes	yes	yes	yes	yes	no	no	no	7/13
65_Yeast _0.0032(37)	no	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	no	11/13
105_Yeast _0.0029	no	no	no	no	yes	yes	yes	yes	yes	no	no	no	no	5/13
52_Random _0.0028	no	yes	no	no	no	yes	yes	yes	yes	no	yes	no	no	6/13
102_Yeast _0.0026	no	yes	no	no	yes	yes	yes	yes	yes	yes	yes	no	yes	9/13
1_Yeast _0.0022	yes	no	yes	yes	no	yes	yes	yes	no	no	no	no	no	6/13
51_Random _0.0021	no	yes	yes	yes	yes	yes	yes	yes	yes	no	no	yes	no	9/13
90_Yeast _0.0021	yes	yes	no	no	no	yes	yes	yes	yes	no	yes	no	no	7/13
3045_Human _0.002	no	yes	no	no	no	yes	yes	no	no	no	no	no	yes	4/13

Continued On Next Page

Table A.2 – *Continued From Previous Page*

Sequence Name	KnotIn Frame	Cylo Fold	IPknot -N	IPknot -P	pknots RG-M	pknots RG-F	pknots RG-L	DotKnot -P	DotKnot -K	MCFold	Kine Fold	Prob Knot	Vsfold5	Scores
2811_Human _0.0018	no	yes	yes	yes	yes	yes	yes	no	no	yes	yes	yes	yes	10/13
15_Random _0.0018	no	yes	no	no	yes	yes	yes	yes	yes	yes	yes	no	yes	9/13
871_Human _0.0017	no	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	no	no	9/13
1951_Human _0.0013	no	yes	yes	yes	no	yes	yes	yes	yes	no	yes	no	no	8/13
643_Human _0.0013	no	yes	no	no	yes	yes	yes	yes	yes	no	no	no	no	6/13
89_Yeast _0.0009(41)	no	no	no	no	no	yes	yes	no	no	yes	no	no	no	3/13
Ratio (yes/49)	17/49	38/49	22/49	22/49	26/49	49/49	49/49	39/49	38/49	21/48	32/49	9/49	21/49	383/636
Ratio_Yeast (yes/16)	6/16	11/16	6/16	6/16	8/16	16/16	16/16	12/16	11/16	8/16	9/16	3/16	4/16	116/208
Ratio_Random (yes/15)	5/15	13/15	8/15	8/15	8/15	15/15	15/15	12/15	12/15	8/15	11/15	4/15	8/15	127/195

Continued On Next Page

Table A.2 – *Continued From Previous Page*

Sequence Name	KnotIn Frame	Cylo Fold	IPknot -N	IPknot -P	pknots RG-M	pknots RG-F	pknots RG-L	DotKnot -P	DotKnot -K	MCFold	Kine Fold	Prob Knot	Vsfold5	Scores
Ratio_Human (yes/18)	6/18	14/18	8/18	8/18	10/18	18/18	18/18	15/18	15/18	5/17	12/18	2/18	9/18	140/233

[1]Whether the particular method has an editable output.

Appendix B

The Comparison of Predicting 34 Viral Frameshifting Signals in PseudoBase

Table B.1: The Comparison Based on the 17 Learning Signals.

Gene Name	**Program**	**Result**
LDV-C (PKB217)	Sequence	UUUAAACUGCUAGCCACCUCUGGUCUCGACCGCUGUACUAGAGGUGGGCUGACGGUGUYUGGCGAUGCGGUCA
	Slippery Site	UUUAAAC
	SubSequence	UGCUAGCCACCUCUGGUCUCGACCGCUGUACUAGAGGUGGGCUGACGGUGUYUGGCGAUGCGGUCA
	PseudoBase	......((((((((((...[[[[[[...))))))))))).................]]]]]].
	Orphea	......((((((((((.....[[[[[[))))))))))..........]]]]]]..........
	KnotInFrame	......((((((((((.....[[[[[[))))))))))....]]]]]]...............
PRRSV-16244B (PKB218)	Sequence	UUUAAACUGCUAGCCGCCAGCGGCUUGACCCGCUGUGGUCGCGGCGGCUUGGUUGUUACUGAGACAGCGGUA
	Slippery Site	UUUAAAC
	SubSequence	UGCUAGCCGCCAGCGGCUUGACCCGCUGUGGUCGCGGCGGCUUGGUUGUUACUGAGACAGCGGUA
	PseudoBase	.....((((((((((((....[[[[[[[))))))))))))).............]]]]]]]..
	Orphea	....(((((((.((((((....[[[[[[[))))))))))))).............]]]]]]]..
	KnotInFrame	....(((((((.(((((.....))))).[[[..)))))))..(((....)))..]]].......
PRRSV -LV (PKB233)	Sequence	UUUAAACUGUUAGCCGCCAGCGGCUUGACCCGCUGUGGCCGCGGCGGCCUAGUUGUGACUGAAACGGCGGU
	Slippery Site	UUUAAAC
	SubSequence	UGUUAGCCGCCAGCGGCUUGACCCGCUGUGGCCGCGGCGGCCUAGUUGUGACUGAAACGGCGGU
	PseudoBase	.....((((((((((((....[[[[[[[))))))))))))).............]]]]]]].
	Orphea	.....((((((.((((((....[[[[[[[))))))))))))).............]]]]]]].
	KnotInFrame	....(((((.[[[)))))......((((.(((((...)))))))))).....]]].....
	Sequence	GGGAAAUGGACUGAGCGGCGCCGACCGCCAAACAACCGGCA

Continued On Next Page

Table B.1 – *Continued From Previous Page*

Gene Name	Program	Result
BChV (PKB240)	Slippery Site	GGGAAAU
	SubSequence	GGACUGAGCGGCGCCGACCGCCAAACAACCGGCA
	PseudoBase	.......((((.[[[[.)))).......]]]].
	Orphea	
	KnotInFrame	No suitable slippery sites have been detected.
BEV (PKB128)	Sequence	UUUAAACUGUUGAGAGGUGCCUGGAGCGCCUGCAGGCAUCUCUGUUUUCAAAAUGGCGCAUACCAGUCUUCAAGGUCAAAACAUUAUAUUGAUUUGGCAACUGAGUAUAAUGCAGGCA
	Slippery Site	UUUAAAC
	SubSequence	UGUUGAGAGGUGCCUGGAGCGCCUGCAGGCAUCUCUGUUUUCAAAAUGGCGCAUACCAGUCUUCAAGGUCAAAACAUUAUAUUGAUUUGGCAACUGAGUAUAAUGCAGGCA
	PseudoBase	.....((((((((((....[[[[[))))))))))).....]]]]].
	Orphea	.....((((((((((..[[[[[..))))))))))..........]]]]]..... ....
	KnotInFrame	.....((((((((((..[[[[[..))))))))))..........]]]]]..... ....
BLV (PKB1)	Sequence	AAAAAACUAAUAGAGGGGGGACUUAGCGCCCCCCAAACCGUAACCCC
	Slippery Site	AAAAAAC
	SubSequence	UAAUAGAGGGGGGACUUAGCGCCCCCCAAACCGUAACCCC
	PseudoBase	.......((((((.....[[[)))))....]]]......

Continued On Next Page

Table B.1 – *Continued From Previous Page*

Gene Name	Program	Result
	Orphea	.......((((((......[[)))))).....]].......
	KnotInFrame	.......((((((..[[[...))))))......]]]....
BWYV (PKB2)	Sequence	GGGAAACGGAGUGCGCGGCACCGUCCGCGGAACAAACGGAGAAGGCAGCU
	Slippery Site	GGGAAAC
	SubSequence	GGAGUGCGCGGCACCGUCCGCGGAACAAACGGAGAAGGCAGCU
	PseudoBase	......(((((..[[[[)))))......]]]]..........
	Orphea	......(((((..[[[[)))))......]]]]..........
	KnotInFrame	......(((((..[[[[)))))......]]]]..........
BYDV-NY-RPV (PKB46)	Sequence	GGGAAACGGGAAGGCGGCGGCGUCCGCCGUAACAAACGC
	Slippery Site	GGGAAAC
	SubSequence	GGGAAGGCGGCGGCGUCCGCCGUAACAAACGC
	PseudoBase	.....(((((..[[[[)))))).......]]]]
	Orphea	.......((((((([[)))))).....]]].
	KnotInFrame	No suitable slippery sites have been detected.
CABYV (PKB44)	Sequence	GGGAAACGGGCAGGCGGCGGCGACCGCCGAAACAACCGC
	Slippery Site	GGGAAAC
	SubSequence	GGGCAGGCGGCGGCGACCGCCGAAACAACCGC
	PseudoBase	.....(((((..[[[.)))))........]]]
	Orphea	.......((((((([.))))))).......]].
	KnotInFrame	No suitable slippery sites have been detected.

Continued On Next Page

Table B.1 – *Continued From Previous Page*

Gene Name	Program	Result
EIAV (PKB3)	Sequence	AAAAAACGGGAAGCAAGGGGCUCAAGGGAGGCCCCAGAAACAAACUUUCCCGAU
	Slippery Site	AAAAAAC
	SubSequence	GGGAAGCAAGGGGCUCAAGGGAGGCCCCAGAAACAAACUUUCCCGAU
	PseudoBase	.....((((((...[[[[))))))............]]]]...
	Orphea	.....((((((...[[[[))))))............]]]]...
	KnotInFrame	No suitable slippery sites have been detected.
FIV (PKB4)	Sequence	GGGAAACUGGAAGGCGGGGCGAGCUGCAGCCCCAGUGAAUCAAAUGCAGC
	Slippery Site	GGGAAAC
	SubSequence	UGGAAGGCGGGGCGAGCUGCAGCCCCAGUGAAUCAAAUGCAGC
	PseudoBase	.....(((((..[[[[[[)))))...........]]]]]]
	Orphea	.....(((((..[[[[[[)))))...........]]]]]]
	KnotInFrame	No suitable slippery sites have been detected.
IBV (PKB106)	Sequence	UUUAAACGGGUACGGGGUAGCAGUGAGGCUCGGCUGAUACCCCUUGCUAGUGGAUGUGAUCCUGAUGUUGUAAAGCGAGCCUU
	Slippery Site	UUUAAAC
	SubSequence	GGGUACGGGGUAGCAGUGAGGCUCGGCUGAUACCCCUUGCUAGUGGAUGUGAUCCUGAUGUUGUAAAGCGAGCCUU
	PseudoBase	......((((((((((.[[[[[[[))))))))))..............................]]]]]].
	Orphea	......((((((.((((.[[[[[[[)))).))))))..............................]]]]]].
	KnotInFrame	.....((((.....[[[[[[[)))).........]]]]]].....
	Sequence	AAAAAACUUGUAAAGGGGCAGUCCCCUAGCCCCGCUCAAAAGGGGGAUG

Continued On Next Page

Table B.1 – *Continued From Previous Page*

Gene Name	Program	Result
MMTV_	Slippery Site	AAAAAAC
gag/pro	SubSequence	UUGUAAAGGGGCAGUCCCCUAGCCCCGCUCAAAAGGGGGAUG
(PKB80)	PseudoBase	.......(((((.[[[[[[.)))))........]]]]]].
	Orphea	.......(((((..[[[[[.)))))........]]]]]]..
	KnotInFrame	.......(((((...[[[[.)))))........]]]]....
	Sequence	GGGAAACGGAUUAUUCCGGUCGACUCCGGAGAAACAAAGUC
	Slippery Site	GGGAAAC
PEMV	SubSequence	GGAUUAUUCCGGUCGACUCCGGAGAAACAAAGUC
(PKB45)	PseudoBase	......((((((..[[[[))))))......]]]]
	Orphea	0 Pseudoknot found.
	KnotInFrame	No suitable slippery sites have been detected.
	Sequence	UUUAAAUGGGCAAGCGGCACCGUCCGCCAAAACAAACGG
	Slippery Site	UUUAAAU
PLRV-S	SubSequence	GGGCAAGCGGCACCGUCCGCCAAAACAAACGG
(PKB43)	PseudoBase	......((((..[[[[))))........]]]]
	Orphea	......((((..[[[[))))........]]]]
	KnotInFrame	No suitable slippery sites have been detected.
	Sequence	UUUAAAUGGGCGAGCGGCACCGCCCGCCAAAACAAACGG
	Slippery Site	UUUAAAU
PLRV-W	SubSequence	GGGCGAGCGGCACCGCCCGCCAAAACAAACGG

Continued On Next Page

Table B.1 – *Continued From Previous Page*

Gene Name	**Program**	**Result**
(PKB42)	PseudoBase	`......((((..[[[.))))........]]]`
	Orphea	`0 Pseudoknot found.`
	KnotInFrame	`No suitable slippery sites have been detected.`
SRV1_ gag/pro (PKB107)	Sequence	`GGGAAACGGACUGAGGGGCCAGCCCCAGGCCCCGAAACAAGCUUAUGGGGCG`
	Slippery Site	`GGGAAAC`
	SubSequence	`GGACUGAGGGGCCAGCCCCAGGCCCCGAAACAAGCUUAUGGGGCG`
	PseudoBase	`.......((((((.[[[[[[))))))............]]]]]].`
	Orphea	`.......((((((.[[[[[[))))))............]]]]]].`
	KnotInFrame	`.......(((((.[[[.....)))))).......]]]........`

Table B.2: The Comparison Based on the 17 Testing Signals.

Gene Name	**Program**	**Result**
WBV (PKB253)	Sequence	UUUAAACUGGUGGGGCAGUGUCUAGGAUUGACGUUAGACACUGCUUUUUGCCCGUUUCAAACAGGUGAAUACAAACCGUCAU
	Slippery Site	UUUAAAC
	SubSequence	UGGUGGGGCAGUGUCUAGGAUUGACGUUAGACACUGCUUUUUGCCCGUUUCAAACAGGUGAAUACAAACCGUCAU
	PseudoBase	....((((((((((((...[[[[[))))))))))))).....]]]]].
	Orphea	.......((((((((((...[[[[[))))))))))).....]]]]].
	KnotInFrame	....((((((((((((....[[[[))))))))))))).....]]]].....
SARS-CoV (PKB254)	Sequence	UUUAAACGGGUUUGCGGUGUAAGUGCAGCCCGUCUUACACCGUGCGGCACAGGCACUAGUACUGAUGUCGUCUACAGGGCUU
	Slippery Site	UUUAAAC
	SubSequence	GGGUUUGCGGUGUAAGUGCAGCCCGUCUUACACCGUGCGGCACAGGCACUAGUACUGAUGUCGUCUACAGGGCUU
	PseudoBase	.....((((((((((...[[[[[[[))))))))))((((((((.........)))).))))))...]].]]]]].
	Orphea	.......(((((((((...[[[[[..)))))))))).....]]]]].
	KnotInFrame	.((((.[[[[[[[[[[...)))))...]]]]]]]]]].....
Mm_Edr (PKB257)	Sequence	GGGAAACUCCCCGGCCCCGCUGUAGGGGGACCUUCAGCGACAGGGCCAGAACGAAUAAGGUCCCCA
	Slippery Site	GGGAAAC
	SubSequence	UCCCCGGCCCCGCUGUAGGGGGACCUUCAGCGACAGGGCCAGAACGAAUAAGGUCCCCA
	PseudoBase	.....((((((((((...[[[[[[[[[)))))...)))))).........]]]]]]]]].
	Orphea	0 PK found.
	KnotInFrame	.((((.....[[[[[..)))).....]]]]].....
	Sequence	GGGAAACGGGAACUGGGCUUGGGACAAGAGCCAUCCCAAGUCCAAGGCCAAGUAGGCUCG

Continued On Next Page

Table B.2 – *Continued From Previous Page*

Gene Name	Program	Result
Hs_Ma3 (PKB258)	Slippery Site	GGGAAAC
	SubSequence	GGGAACUGGGCUUGGGACAAGAGCCAUCCCAAGUCCAAGGCCAAGUAGGCUCG
	PseudoBase	......((((((((((...[[[[[.))))))))))).........]]]]].
	Orphea	......((((((((((...[[[[[.))))))))))).........]]]]].
	KnotInFrame	........(((((.[[[[..))))).......]]]].................
VMV (PKB280)	Sequence	GGGAAACAACAGGAGGGGGCCACGUGUGGUGCCGUCCGCGCCCCCUAUGUUGUAACAGAAGCACCACC
	Slippery Site	GGGAAAC
	SubSequence	AACAGGAGGGGGCCACGUGUGGUGCCGUCCGCGCCCCCUAUGUUGUAACAGAAGCACCACC
	PseudoBase	......(((((((.....[[[[[[[.......)))))))).............]]]]]]].
	Orphea	......(((((((.....[[[[[[[.......)))))))).............]]]]]]].
	KnotInFrame	......(((((((......[[[[[[.......))))))).(((....)))...]]]]]]..
ScYLV (PKB281)	Sequence	GGGAAACGAGCCAAGUGGCGCCGACCACUUAAAAACACCGGAA
	Slippery Site	GGGAAAC
	SubSequence	GAGCCAAGUGGCGCCGACCACUUAAAAACACCGGAA
	PseudoBase	......(((((..[[[.)))))).........]]]..
	Orphea	.....((((((..[[[.))))))........]]]..
	KnotInFrame	No suitable slippery sites have been detected.
KUNV	Sequence	UCCUUUUCAGCUGGGCCUUCUGGUCGUGUUCUUGGCCACCCAGGAGGUCCUUCGCAAGAGGUGGACAGCCAAGAU
	Slippery Site	UCCUUUU
	SubSequence	CAGCUGGGCCUUCUGGUCGUGUUCUUGGCCACCCAGGAGGUCCUUCGCAAGAGGUGGACAGCCAAGAU

Continued On Next Page

Table B.2 – *Continued From Previous Page*

Gene Name	Program	Result
(PKB346)	PseudoBase	.....((((((((((.....[[[[[[[...)))))))))).((....)).....]]]]]].
	Orphea	0 PK found.
	KnotInFrame	No suitable slippery sites have been detected.
WNV (PKB347)	Sequence	CCCUUUUCAGUUGGGCCUUCUGGUCGUGUUCUUGGCCACCCAGGAGGUCCUUCGCAAGAGGUGGACAGCCAAGAU
	Slippery Site	CCCUUUU
	SubSequence	CAGUUGGGCCUUCUGGUCGUGUUCUUGGCCACCCAGGAGGUCCUUCGCAAGAGGUGGACAGCCAAGAU
	PseudoBase	.....((((((((((.....[[[[[[[...)))))))))).((....)).....]]]]]].
	Orphea	.....((((((((((.....[[[[[[[...)))))))))).............]]]]]].
	KnotInFrame	.....((((((((((.((.....)).[[[[)))))))))).........]]]]..........
JEV (PKB348)	Sequence	CCCUUUUCAGCUGGGCCUUCUGGUGAUGUUUCUGGCCACCCAGGAGGUCCUUCGCAAGAGGUGGACGGCCAGAUUGA
	Slippery Site	CCCUUUU
	SubSequence	CAGCUGGGCCUUCUGGUGAUGUUUCUGGCCACCCAGGAGGUCCUUCGCAAGAGGUGGACGGCCAGAUUGA
	PseudoBase	.....((((((((((.....[[[[[[[..)))))))))).((....)).....]]]]]]]....
	Orphea	.....((((((((((.....[[[[[[[..)))))))))).............]]]]]]]....
	KnotInFrame	.....((((((((((.((....))...[[[[)))))))))).........]]]]..........
MVEV (PKB349)	Sequence	UCCUUUUCAGUUAGGCCUUCUGGUGAUGUUUCUGGCCACCCAGGAGGUCUUGAGGAAGAGGUGGACGGCCAGACUUACUC
	Slippery Site	UCCUUUU
	SubSequence	CAGUUAGGCCUUCUGGUGAUGUUUCUGGCCACCCAGGAGGUCUUGAGGAAGAGGUGGACGGCCAGACUUACUC
	PseudoBase	.....((((((((((.....[[[[[[[..)))))))))).............]]]]]]].....
	Orphea	0 PK found.

Continued On Next Page

Table B.2 – *Continued From Previous Page*

Gene Name	Program	Result
	KnotInFrame	No suitable slippery sites have been detected.
ALFV (PKB350)	Sequence	CCCUUUUCAGCUGGGCCUCUUGGUAGUUUUCCUGGCCACCCAGGAGGUCUUGAGGAAGAGGUGGACGGCCAGAAUGA
	Slippery Site	CCCUUUU
	SubSequence	CAGCUGGGCCUCUUGGUAGUUUUCCUGGCCACCCAGGAGGUCUUGAGGAAGAGGUGGACGGCCAGAAUGA
	PseudoBase	.....((((((((((........[[[[[[..))))))))))).............]]]]]].....
	Orphea	......((((((((((........[[[[[[..))))))))))).............]]]]]].....
	KnotInFrame	..((((((.[[[[[[[.......))))))...]]]]]]]...........................
USUV (PKB351)	Sequence	AUCCUUUUCAGUUGGGCCUUCUGGUGAUGUUUCUGGCCACCCAGGAGGUCCUGAGGAAGAGGUGGACGGCCAGAUUGACU
	Slippery Site	UCCUUUU
	SubSequence	CAGUUGGGCCUUCUGGUGAUGUUUCUGGCCACCCAGGAGGUCCUGAGGAAGAGGUGGACGGCCAGAUUGACU
	PseudoBase	.....((((((((((.......[[[[[[[..))))))))))).............]]]]]]]......
	Orphea	0 PK found.
	KnotInFrame	No suitable slippery sites have been detected.
MIDV (PKB352)	Sequence	UUCUUUUUUAGUGGCAGUAAGCCUGGGAAUGGGGGGCGACCCAGGCGUAUGAACAUAGUGUAACGCUCCCC
	Slippery Site	UUUUUUA
	SubSequence	GUGGCAGUAAGCCUGGGAAUGGGGGCGACCCAGGCGUAUGAACAUAGUGUAACGCUCCCC
	PseudoBase	......(((.(((((((...[[[[[[[.))))))).))).............]]]]]]].
	Orphea	0 PK found.
	KnotInFrame	..........(((((((...[[[[[[[.))))))).....(((...))).]]]]]]].
	Sequence	UUUGUUUUUUAGCUGUGCUGGGUGCGAGUGUGGCAGCGGCUCGUGCCUACGAACACACCGCUGUCAUGCC

Continued On Next Page

Table B.2 – *Continued From Previous Page*

Gene Name	Program	Result
SESV (PKB353)	Slippery Site	UUUUUUA
	SubSequence	GCUGUGCUGGGUGCGAGUGUGGCAGCGGCUCGUGCCUACGAACACACCGCUGUCAUGCC
	PseudoBase	.......((((((((((.[[[[[[[[)))))))))).........]]]]]]]]....
	Orphea	.......((((((((((.[[[[[[[[)))))))))).........]]]]]]]]....
	KnotInFrame	.((((..[[[[[[[[[[...)))))..]]]]]]]]]]....................
EAV (PKB127)	Sequence	GUUAAACUGAGAGCGCCCCACAUCUUUCCCGGCGAUGUGGGGCGUCGGACCUUUGCUGACUCUAAAGACAAGGGUUUCGUGGCUCUACACAGU CGCACAAUGUUUUUAGCUGCCCGGGACUU
	Slippery Site	GUUAAAC
	SubSequence	UGAGAGCGCCCCACAUCUUUCCCGGCGAUGUGGGGCGUCGGACCUUUGCUGACUCUAAAGACAAGGGUUUCGUGGCUCUACACAGUCGCACAA UGUUUUUAGCUGCCCGGGACUU
	PseudoBase	......((((((((((..[[[[[[.)))))))))).....]]]]]]...
	Orphea	......((((((((((...[[[...)))))))))).....]]].....
	KnotInFrame	No suitable slippery sites have been detected.
RSV	Sequence	AAAUUUAUAGGGAGGGCCACUGUUCUCACUGUUGCGCUACAUCUGGCUAUUCCGCUCAAAUGGAAGCCAGACCACACGCCUGUGUGGAUUGAC CAGUGGCCCCUCCCUGAAGGUAAACUUGUAGCGCU
	Slippery Site	AAAUUUA
	SubSequence	UAGGGAGGGCCACUGUUCUCACUGUUGCGCUACAUCUGGCUAUUCCGCUCAAAUGGAAGCCAGACCACACGCCUGUGUGGAUUGACCAGUGGC CCCUCCCUGAAGGUAAACUUGUAGCGCU

Continued On Next Page

Table B.2 – *Continued From Previous Page*

Gene Name	Program	Result
(PKB174)	PseudoBase	.((((((((((((((...........[[[[[[[[((((((..((((((.....))))))))))))((((((....))))))).....))))))))).)))))...........]]]]]]]].
	Orphea	0 PK found.
	KnotInFrame	......(((((((((.............[[[[[[[[(((((((..((((.....))))))))))))((((((....))))))).....)))))))))..((.....)).....]]]]]]]..
HCV_ 229E	Sequence	UUUAAACGAGUCCGGGGCUCUAGUGCCGCUCGACUAGAGCCCUGUAAUGGUACAGACAUAGAUUACUGUGUCCGUGCAUUUGACGUUUACAAU AAAGAUGCGUCUUUUAUCGGAAAAAAAUCUGAAGUCCAAUUGUGUGCGCUUCAAGAAUGUAGAUAAGGAUGACGCGUUCUAUAUUGUUAAACGU UGCAUUAAGUCAGUUAUGGACCACGAGCAGUCCAUGUA
	Slippery Site	UUUAAAC
	SubSequence	GAGUCCGGGGCUCUAGUGCCGCUCGACUAGAGCCCUGUAAUGGUACAGACAUAGAUUACUGUGUCCGUGCAUUUGACGUUUACAAUAAAGAUG CGUCUUUUAUCGGAAAAAAAUCUGAAGUCCAAUUGUGUGCGCUUCAAGAAUGUAGAUAAGGAUGACGCGUUCUAUAUUGUUAAACGUUGCAUUA AGUCAGUUAUGGACCACGAGCAGUCCAUGUA
(PKB171)	PseudoBase	.....((((((((((((...[[[[[)))))))))))).......... .. ..(((((((..]]]]].))))))))..
	Orphea	.....((((((((((((.[[[....))))))))))))...]]]..... .. ....

Continued On Next Page

Table B.2 – *Continued From Previous Page*

Gene Name	Program	Result
	KnotInFrame	.....((((((((((((.[[[....))))))))))))).....((((.(((((((....)))))))).))))....(((((.....)))))))))......]]].....

Appendix C

The Classification of the 414 sequences in the Benchmark

Table C.1: The Classification of the 414 sequences in the Benchmark

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB1	47	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB2	50	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB3	54	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB4	50	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB5	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB6	42	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB7	42	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB8	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB9	42	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB10	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB11	42	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB12	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB13	42	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB14	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB15	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB16	42	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB17	42	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB18	38	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB19	38	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB20	38	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB21	38	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB22	39	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB23	38	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB24	39	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB25	37	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB26	37	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB27	38	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB28	37	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB29	37	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB30	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB31	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB32	38	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB33	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB34	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB35	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB36	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB37	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB38	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB39	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB40	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB41	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB42	39	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB43	39	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB44	39	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB45	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB46	39	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB47	61	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB48	61	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB49	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB50	59	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB51	46	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB52	52	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB53	22	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB54	22	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB55	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB56	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB57	67	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB58	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB59	22	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB60	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB61	22	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB62	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB63	22	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB64	920	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB65	46	simple H-type	ABAB	2	N	Y	Y	Y	Y
PKB66	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB67	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB68	68	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB69	61	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB70	55	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB71	108	complex	ABCABC	3	N	N	N	N	Y
PKB72	67	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB73	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB74	28	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – *Continued From Previous Page*

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB75	88	complex	ABCDCADB	2	N	N	N	N	Y
PKB76	89	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB77	219	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB78	62	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB79	61	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB80	49	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB81	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB82	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB83	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB84	32	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB85	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB86	22	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB87	32	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB88	62	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB89	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB90	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB91	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB92	27	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB93	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB94	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB95	23	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB96	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB97	32	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB98	62	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB99	63	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB100	31	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB101	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB102	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB103	25	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB104	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB105	32	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB106	83	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB107	52	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB108	35	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB109	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB110	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB111	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB112	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB113	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB114	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB115	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB116	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB117	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB118	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB119	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB120	36	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB121	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB122	31	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB123	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB124	29	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB125	25	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB126	27	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB127	122	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB128	118	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB129	313	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB130	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB131	48	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB132	49	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB133	48	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB134	137	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB135	116	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB136	134	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB137	133	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB138	96	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB139	70	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB140	69	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB141	70	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB142	71	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB143	71	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB144	71	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB145	58	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB146	50	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB147	51	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB148	108	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB149	351	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB150	212	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB151	32	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB152	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB153	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB154	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB155	21	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB156	23	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB157	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB158	28	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB159	25	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB160	32	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB161	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB162	35	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB163	47	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
PKB164	96	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB165	23	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB166	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB167	35	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB168	105	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB169	73	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB170	149	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB171	224	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
PKB172	39	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB173	73	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
PKB174	128	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB175	57	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB176	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB177	70	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB178	90	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
PKB179	124	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB180	143	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB181	207	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB182	42	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB183	27	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB184	31	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB185	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB186	29	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB187	27	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB188	23	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB189	28	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB190	47	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB191	113	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB193	341	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB194	28	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB195	31	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB196	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB197	29	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB198	32	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB199	23	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB200	28	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB201	29	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB202	34	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB203	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB204	29	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB205	48	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB206	45	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB207	45	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB208	237	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – *Continued From Previous Page*

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB209	234	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB210	90	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB211	146	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB212	64	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB213	45	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB214	145	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB215	64	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB216	45	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB217	73	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB218	72	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB219	147	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB220	64	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB221	45	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB222	146	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB223	64	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB224	43	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB225	147	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB226	64	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB227	44	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – *Continued From Previous Page*

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB228	148	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB229	67	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB230	48	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB231	130	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB232	62	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB233	71	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB234	84	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB235	77	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB236	120	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB237	96	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB238	84	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB239	412	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB240	41	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB241	34	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB242	34	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB243	121	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB244	55	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB245	35	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB246	34	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – *Continued From Previous Page*

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB247	22	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB248	66	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB249	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB250	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB251	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB252	110	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB253	82	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB254	82	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB255	56	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB256	56	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB257	66	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB258	60	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB259	57	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB260	57	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB261	59	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB262	56	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB263	62	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB264	62	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB265	61	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB266	47	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB267	72	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB268	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB269	66	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB270	62	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB271	75	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB272	66	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB273	48	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB274	49	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB275	85	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB276	73	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB277	37	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB278	29	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB279	21	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB280	68	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB281	43	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB282	28	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB283	29	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB284	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB285	27	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB286	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB287	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB288	25	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB289	28	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB290	30	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB291	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB292	25	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB293	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB294	25	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB295	24	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB296	26	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB297	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB298	29	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB299	25	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB300	37	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB301	37	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB302	31	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB303	76	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB304	34	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB305	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB306	78	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB307	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB308	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB309	145	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB310	130	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB311	120	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB312	130	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB313	130	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB314	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB315	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB316	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB317	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB318	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB319	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB320	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB321	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB322	67	complex	ABCDCADB	2	N	N	N	N	Y

Continued On Next Page

Table C.1 – *Continued From Previous Page*

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB323	180	complex	ABCDCADB	2	N	N	N	N	Y
PKB324	181	complex	ABCDCADB	2	N	N	N	N	Y
PKB325	78	complex	ABCDCADB	2	N	N	N	N	Y
PKB326	63	complex	ABCDCADB	2	N	N	N	N	Y
PKB327	82	complex	ABCDCADB	2	N	N	N	N	Y
PKB328	81	complex	ABCDCADB	2	N	N	N	N	Y
PKB329	82	complex	ABCDCADB	2	N	N	N	N	Y
PKB330	64	complex	ABCDCADB	2	N	N	N	N	Y
PKB331	64	complex	ABCDCADB	2	N	N	N	N	Y
PKB332	68	complex	ABCDCADB	2	N	N	N	N	Y
PKB333	68	complex	ABCDCADB	2	N	N	N	N	Y
PKB334	77	complex	ABCDCADB	2	N	N	N	N	Y
PKB335	104	complex	ABCDCADB	2	N	N	N	N	Y
PKB336	106	complex	ABCDCADB	2	N	N	N	N	Y
PKB337	106	complex	ABCDCADB	2	N	N	N	N	Y
PKB338	66	complex	ABCDCADB	2	N	N	N	N	Y
PKB339	69	complex	ABCDCADB	2	N	N	N	N	Y
PKB340	67	complex	ABCDCADB	2	N	N	N	N	Y
PKB341	76	complex	ABCDCADB	2	N	N	N	N	Y

Continued On Next Page

Table C.1 – *Continued From Previous Page*

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB342	88	complex	ABCDCADB	2	N	N	N	N	Y
PKB343	54	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB344	94	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
PKB345	52	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB346	75	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB347	75	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB348	77	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB349	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB350	77	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB351	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB352	70	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB353	70	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB354	190	complex	ABCDCADB	2	N	N	N	N	Y
PKB355	150	complex	ABCDCADB	2	N	N	N	N	Y
PKB356	140	complex	ABCDCADB	2	N	N	N	N	Y
PKB357	160	complex	ABCDCADB	2	N	N	N	N	Y
PKB358	190	complex	ABCDCADB	2	N	N	N	N	Y
PKB359	40	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB360	130	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – ***Continued From Previous Page***

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
PKB361	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB362	90	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB363	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB364	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB365	90	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB366	80	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB367	64	simple H-type	ABAB	2	Y	Y	Y	Y	Y
PKB192	1248	simple H-type	ABAB	2	Y	Y	Y	Y	Y
2KFC_A	36	simple H-type	ABAB	2	Y	Y	Y	Y	Y
2KRL_A	102	simple H-type	ABAB	2	Y	Y	Y	Y	Y
2LC8_A	56	simple H-type	ABAB	2	Y	Y	Y	Y	Y
2M58_A	58	simple H-type	ABAB	2	Y	Y	Y	Y	Y
2RP0_A	27	simple H-type	ABAB	2	Y	Y	Y	Y	Y
2WDL_A	2807	complex	others	4	N	N	N	N	N
2ZZN_D	71	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3A2K_C	77	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3A3A_A	86	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3ADB_C	92	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3GCA_A	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y

Continued On Next Page

Table C.1 – *Continued From Previous Page*

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
3GX2_A	94	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3IVN_B	69	complex	ABCBDADECE	2	N	N	N	N	Y
3IWN_A	93	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3IYQ_A	349	recursive H-type	ABAB	2	N	Y	Y	Y	Y
3IZ4_A	377	recursive H-type	ABAB	2	N	Y	Y	Y	Y
3J0L_A	48	simple H-type	ABAB	2	N	Y	Y	Y	Y
3J3D_C	75	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3J3E_8	123	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3J3F_8	157	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3J20_0	76	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3J20_2	1495	complex	others	3	N	N	N	N	Y
3JYV_7	76	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3KIY_A	2848	complex	others	4	N	N	N	N	N
3LA5_A	71	complex	ABCBDADECE	2	N	N	N	N	Y
3NKB_B	64	simple H-type	ABAB	2	N	Y	Y	Y	Y
3NPB_A	119	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3O58_3	158	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3PDR_A	161	simple H-type	ABAB	2	N	Y	Y	Y	Y
3RKF_A	67	complex	ABCBDADECE	2	N	N	N	N	Y

Continued On Next Page

Table C.1 – *Continued From Previous Page*

PKBNo.	Length	Pseudoknot Type	Pseudobase Pattern	Page No.	L&P class	D&P class	A&U class	J&C class	R&E class
3SD1_A	89	simple H-type	ABAB	2	N	Y	Y	Y	Y
3U4M_B	80	simple H-type	ABAB	2	N	Y	Y	Y	Y
3W1K_J	92	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3W3S_B	98	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
3ZEX_C	169	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
4A1C_2	154	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
4AOB_A	94	Kissing Hairpin	ABACBC	2	N	N	N	Y	Y
4ATO_G	33	simple H-type	ABAB	2	Y	Y	Y	Y	Y
4ENB_A	51	simple H-Type	ABAB	2	Y	Y	Y	Y	Y
4ENC_A	52	simple H-Type	ABAB	2	Y	Y	Y	Y	Y
4FRG_B	84	simple H-type	ABAB	2	N	Y	Y	Y	Y
4FRN_A	102	complex	ABCDCEBEAFDF	2	N	N	N	N	Y
4JF2_A	76	simple H-type	ABAB	2	Y	Y	Y	Y	Y
4JRC_A	56	simple H-type	ABAB	2	N	Y	Y	Y	Y
3J2C_N	927	recursive H-type	ABAababcdcdB	2	N	Y	Y	Y	Y
3JYX_5	3170	recursive Kissing Hairpin	ababABAcdcdDEDFEFefefCBC	2	N	N	N	Y	Y
3ZEX_B	1465	complex	others	3	N	N	N	N	N

Appendix D

The Evaluation Values of the Prediction by the 15 Methods

There are 15 methods considered in the benchmark, twelve heuristic methods and three exact ones. Typically, the heuristic methods are listed before the exact methods in the following tables. And each table highlights the winner program for the subsets in bold, which have obtained the highest corresponding evaluation values. And in the end, the *Winner Times* counts the number of obtaining the best evaluation values for each program.

Table D.1: The sensitivity of the predictions.

Attribute	Value	Size	CyloFold	DotKnot	DotKnot-K	HotKnots-cc	HotKnots-dp	HotKnots-re	IPknot	MC-Fold	McGenus	McQFold	pKiss	vsfold5	pknotsRG-M	pknotsRG-F	pknots
All	Entire Set	414	0.737	0.751	0.739	0.681	0.568	0.621	0.578	0.701	0.739	0.632	**0.772**	0.525	0.736	0.743	0.735
	Shared Set	387	0.753	0.766	0.754	0.688	0.567	0.621	0.575	0.701	0.745	0.639	**0.774**	0.535	0.749	0.756	0.737
	Missing Set	27	0.404	0.531	0.52	0.552	0.583	0.615	0.618	0	0.634	0.521	**0.663**	0.316	0.506	0.515	0.634
Length	$\leq$ 100 nt	345	0.773	0.792	0.781	0.705	0.57	0.629	0.577	0.714	0.751	0.641	**0.799**	0.557	0.775	0.78	0.754
	101-160 nt	42	0.592	0.55	0.537	0.541	0.547	0.558	0.56	0.591	**0.688**	0.622	0.57	0.353	0.543	0.559	0.598
RNA Type	Aptamers	15	**0.854**	0.777	0.777	0.717	0.62	0.7	0.657	0.645	0.811	0.754	0.76	0.616	0.749	0.772	0.761
	mRNA	16	0.544	0.46	0.517	0.442	0.375	0.424	0.361	0.436	**0.575**	0.493	0.427	0.374	0.441	0.42	0.559
	tRNA	8	0.767	0.796	**0.854**	0.624	0.624	0.624	0.797	0.566	0.656	0.843	0.739	0.82	0.663	0.618	0.825
	tmRNA	10	0.562	0.653	0.653	0.61	0.594	0.576	0.559	0.716	**0.732**	0.494	0.507	0.324	0.551	0.472	0.54
	rRNA	10	0.443	0.386	0.361	0.432	0.432	0.445	0.41	0.277	0.45	0.44	**0.463**	0.42	0.423	0.336	0.418
	Ribozymes	37	0.756	0.773	0.766	0.669	0.615	0.756	0.62	0.755	0.794	0.775	**0.825**	0.542	0.784	0.802	0.773
	Riboswitch	11	0.751	0.774	0.657	**0.82**	**0.82**	**0.82**	0.749	0.679	0.789	0.777	0.733	0.621	0.775	0.707	0.744
	Others	13	0.691	0.581	0.64	0.632	0.593	0.622	0.549	0.6	**0.75**	0.492	0.697	0.521	0.667	0.62	0.634
	Vr. 3 UTR	103	0.833	0.848	0.851	0.726	0.603	0.698	0.633	0.78	0.802	0.634	**0.923**	0.648	0.868	0.888	0.824
	Vr. 5 UTR	29	0.791	0.846	0.81	0.743	0.523	0.612	0.586	0.698	0.685	0.527	**0.92**	0.632	0.857	0.893	0.775
	Frameshifting	33	0.832	0.799	0.795	0.797	0.571	0.656	0.485	0.775	0.844	0.744	**0.846**	0.287	0.76	0.753	0.772
	Vr. ReadThrough	7	0.666	0.738	0.738	0.49	0.071	0.071	0.071	0.8	**0.862**	0.776	0.427	0.261	0.529	0.676	0.819
	Vr. tRNA-like	58	0.704	**0.822**	0.753	0.692	0.538	0.484	0.546	0.663	0.624	0.596	0.693	0.5	0.753	0.774	0.652

Continued On Next Page

Table D.1 – *Continued From Previous Page*

Attribute	Value	Size	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfold5	pknots RG-M	pknots RG-F	pknots
	Vr. Others	30	0.712	0.651	0.652	0.614	0.575	0.598	0.606	0.698	**0.818**	0.619	0.654	0.487	0.603	0.602	0.746
	Unknown	7	0.779	0.767	0.755	**0.903**	0.609	0.73	0.582	0.81	0.686	0.648	0.786	0.592	0.811	0.832	0.717
Organism	Eukaryote	59	0.613	0.581	0.613	0.548	0.497	0.573	0.515	0.55	**0.659**	0.596	0.626	0.447	0.604	0.603	0.63
	Prokaryote	40	0.727	0.709	0.685	0.67	0.662	0.677	0.638	0.703	**0.75**	0.694	0.688	0.544	0.668	0.617	0.667
	Virus	266	0.78	0.814	0.795	0.708	0.558	0.611	0.57	0.734	0.758	0.63	**0.818**	0.539	0.788	0.809	0.766
	Unknown	22	0.849	0.789	0.76	**0.853**	0.687	0.77	0.685	0.7	0.805	0.754	0.791	0.701	0.825	0.776	0.812
Page No.	2	386	0.754	0.767	0.755	0.688	0.567	0.622	0.576	0.702	0.744	0.639	**0.775**	0.536	0.75	0.757	0.738
	3	1	0.458	0.458	0.458	0.458	0.458	0.458	0.5	0.292	**0.792**	0.458	0.458	0.167	0.458	0.458	0.458
Pknot Type	H-type	330	0.759	0.775	0.76	0.693	0.557	0.609	0.565	0.709	0.747	0.624	**0.777**	0.53	0.754	0.764	0.737
	Kissing	22	0.65	0.602	**0.673**	0.595	0.595	0.599	0.658	0.506	0.587	0.656	0.667	0.628	0.602	0.593	0.653
	Complex	35	0.763	0.783	0.754	0.694	0.64	0.752	0.625	0.743	**0.821**	0.768	0.811	0.518	0.802	0.781	0.79
Winner Time(28)			1	1	2	3	1	1	0	0	10	0	**12**	0	0	0	0

Table D.2: The PPV of the predictions.

Attribute	Value	Size	CyloFold	DotKnot	DotKnot-K	HotKnots-cc	HotKnots-dp	HotKnots-re	IPknot	MC-Fold	McGenus	McQFold	pKiss	vsfold5	pknotsRG-MCM	pknotsRG-MCF	pknots
All	Entire Set	414	0.717	0.723	0.708	0.689	0.633	0.658	0.674	0.607	0.706	0.703	**0.732**	0.584	0.713	0.698	0.708
	Shared Set	387	0.741	**0.752**	0.736	0.71	0.65	0.675	0.694	0.607	0.726	0.727	0.736	0.603	0.737	0.721	0.715
	Missing Set	27	0.236	0.313	0.305	0.279	0.301	0.318	0.398	0	0.298	0.279	**0.535**	0.196	0.292	0.284	0.387
Length	$\leq$ 100 nt	345	0.774	**0.789**	0.774	0.744	0.676	0.703	0.72	0.635	0.754	0.755	0.773	0.641	0.773	0.755	0.748
	101-160 nt	42	0.469	0.443	0.427	0.428	0.438	0.45	0.477	0.379	**0.498**	0.495	0.429	0.296	0.441	0.444	0.444
RNA Type	Aptamers	15	0.898	0.784	0.784	0.759	0.743	0.77	0.84	0.598	0.848	**0.918**	0.76	0.718	0.778	0.771	0.799
	mRNA	16	0.453	0.381	0.429	0.38	0.379	0.394	0.391	0.324	**0.461**	0.453	0.344	0.358	0.369	0.352	0.45
	tRNA	8	0.755	0.772	0.778	0.609	0.609	0.609	0.78	0.461	0.627	**0.863**	0.743	0.838	0.641	0.586	0.807
	tmRNA	10	0.661	0.748	0.748	0.726	**0.783**	0.772	0.734	0.644	0.765	0.657	0.593	0.402	0.735	0.543	0.602
	rRNA	10	0.347	0.305	0.274	0.313	0.313	0.318	0.339	0.18	0.314	**0.373**	0.343	0.311	0.328	0.247	0.305
	Riboswitch	11	0.81	0.827	0.659	0.87	**0.881**	0.862	0.869	0.587	0.812	0.837	0.754	0.644	0.816	0.735	0.745
	Ribozymes	37	0.805	0.83	0.817	0.691	0.626	0.761	0.769	0.682	0.795	0.817	**0.846**	0.632	0.808	0.819	0.775
	Others	13	0.605	0.547	0.585	0.621	0.573	0.621	0.595	0.52	0.636	0.535	0.646	0.57	**0.655**	0.598	0.596
	Vr. 3 UTR	103	0.884	0.89	0.892	0.877	0.853	0.866	0.88	0.787	0.901	0.888	**0.914**	0.814	0.895	0.885	0.896
	Vr. 5 UTR	29	0.854	0.909	0.869	0.872	0.748	0.784	0.75	0.709	0.779	0.74	**0.952**	0.821	0.915	0.909	0.856
	Frameshifting	33	**0.747**	0.708	0.699	0.678	0.499	0.563	0.49	0.504	0.696	0.668	0.739	0.258	0.681	0.683	0.618
	Vr. ReadThrough	7	0.519	**0.692**	0.664	0.426	0.053	0.053	0.053	0.47	0.601	0.6	0.371	0.257	0.487	0.62	0.567
	Vr. tRNA-like	58	0.585	**0.657**	0.609	0.572	0.493	0.471	0.568	0.464	0.513	0.572	0.552	0.464	0.618	0.613	0.535

Continued On Next Page

Table D.2 – *Continued From Previous Page*

Attribute	Value	Size	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfo ld5	pknots RG-M	pknots RG-F	pkn ots
	Vr. Others	30	0.67	0.665	0.658	0.628	0.622	0.63	0.671	0.577	**0.719**	0.638	0.618	0.539	0.627	0.592	0.676
	Unknown	7	0.725	0.727	0.718	**0.834**	0.641	0.685	0.68	0.662	0.648	0.733	0.732	0.617	0.746	0.77	0.674
Organism	Eukaryote	59	0.612	0.596	0.614	0.549	0.498	0.57	0.61	0.473	**0.632**	0.613	0.619	0.497	0.612	0.601	0.602
	Prokaryote	40	0.73	0.707	0.671	0.672	0.682	0.695	0.705	0.591	0.699	**0.738**	0.671	0.55	0.688	0.597	0.654
	Virus	266	0.762	**0.789**	0.773	0.739	0.667	0.684	0.698	0.638	0.745	0.737	0.768	0.619	0.767	0.763	0.743
	Unknown	22	0.845	0.791	0.748	0.859	0.79	0.816	0.839	0.626	0.799	**0.89**	0.777	0.796	0.802	0.761	0.799
Page No.	2	386	0.742	**0.753**	0.738	0.711	0.651	0.676	0.694	0.608	0.727	0.728	0.737	0.605	0.738	0.722	0.716
	3	1	0.367	0.314	0.314	0.297	0.297	0.297	0.333	0.2	**0.528**	0.314	0.289	0.154	0.289	0.306	0.333
Pknot Type	H-Type	330	0.739	**0.753**	0.738	0.717	0.653	0.67	0.684	0.611	0.726	0.721	0.731	0.601	0.736	0.722	0.712
	Kissing	22	0.64	0.563	0.601	0.575	0.575	0.584	**0.662**	0.414	0.543	0.646	0.633	0.633	0.591	0.542	0.613
	Complex	35	0.823	**0.854**	0.809	0.729	0.669	0.777	0.8	0.689	0.837	0.834	0.847	0.61	0.841	0.821	0.809
Winner Time(28)			1	**8**	0	1	2	0	1	0	5	5	5	0	1	0	0

Table D.3: The MCC of the predictions.

Attribute	Value	Size	CyloFold	DotKnot	DotKnot-K	HotKnots-cc	HotKnots-dp	HotKnots-re	IPknot	MC-Fold	McGenus	McQFold	pKiss	vsfold5	pknotsRG-MCM	pknotsRG-FCM	pknots
All	Entire Set	414	0.713	0.723	0.71	0.668	0.578	0.62	0.604	0.633	0.705	0.645	**0.741**	0.535	0.711	0.707	0.706
	Shared Set	387	0.733	**0.746**	0.733	0.683	0.586	0.63	0.613	0.633	0.719	0.662	0.743	0.55	0.731	0.726	0.712
	Missing Set	27	0.295	0.389	0.379	0.376	0.401	0.423	0.466	0	0.415	0.361	**0.582**	0.235	0.367	0.362	0.477
Length	$\leq$ 100 nt	345	0.76	**0.779**	0.765	0.709	0.6	0.647	0.626	0.654	0.737	0.676	0.775	0.579	0.761	0.756	0.737
	101-160 nt	42	0.515	0.482	0.467	0.468	0.477	0.489	0.507	0.461	**0.572**	0.543	0.482	0.311	0.478	0.486	0.502
RNA Type	Aptamers	15	**0.869**	0.771	0.771	0.728	0.664	0.722	0.73	0.602	0.819	0.819	0.751	0.652	0.753	0.762	0.771
	mRNA	16	0.488	0.407	0.461	0.398	0.36	0.395	0.36	0.359	**0.503**	0.458	0.371	0.354	0.392	0.373	0.491
	tRNA	8	0.755	0.779	0.81	0.608	0.608	0.608	0.783	0.499	0.633	**0.849**	0.735	0.825	0.644	0.593	0.812
	tmRNA	10	0.595	0.689	0.689	0.652	0.669	0.654	0.627	0.667	**0.737**	0.554	0.534	0.338	0.623	0.491	0.554
	rRNA	10	0.377	0.325	0.296	0.352	0.352	0.361	0.358	0.203	0.361	**0.391**	0.381	0.344	0.357	0.269	0.34
	Ribozymes	37	0.772	0.794	0.784	0.669	0.608	0.75	0.679	0.708	0.788	0.788	**0.83**	0.574	0.79	0.804	0.765
	Riboswitch	11	0.773	0.794	0.649	0.839	**0.845**	0.835	0.8	0.618	0.794	0.8	0.736	0.623	0.789	0.713	0.737
	Others	13	0.635	0.55	0.601	0.614	0.565	0.607	0.559	0.543	**0.677**	0.495	0.659	0.529	0.649	0.595	0.6
	Vr. 3 UTR	103	0.844	0.857	0.86	0.782	0.698	0.758	0.728	0.766	0.834	0.73	**0.912**	0.706	0.87	0.876	0.846
	Vr. 5 UTR	29	0.809	0.866	0.827	0.788	0.601	0.67	0.642	0.677	0.711	0.598	**0.929**	0.699	0.873	0.892	0.798
	Frameshifting	33	0.777	0.742	0.735	0.725	0.516	0.593	0.47	0.61	0.756	0.691	**0.781**	0.253	0.709	0.707	0.677
	Vr. ReadThrough	7	0.577	0.707	0.691	0.444	0.039	0.039	0.042	0.602	**0.712**	0.675	0.382	0.243	0.495	0.638	0.674
	Vr. tRNA-like	58	0.625	**0.722**	0.662	0.612	0.495	0.458	0.537	0.534	0.548	0.568	0.602	0.463	0.668	0.675	0.575

Continued On Next Page

Table D.3 – *Continued From Previous Page*

Attribute	Value	Size	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfo ld5	pknots RG-M	pknots RG-F	pkn ots
	Vr. Others	30	0.675	0.643	0.639	0.604	0.579	0.595	0.619	0.616	**0.748**	0.605	0.621	0.496	0.599	0.582	0.696
	Unknown	7	0.742	0.736	0.726	**0.863**	0.609	0.697	0.611	0.721	0.654	0.675	0.75	0.588	0.77	0.793	0.684
Organism	Eukaryote	59	0.601	0.577	0.602	0.535	0.483	0.558	0.546	0.496	**0.633**	0.592	0.611	0.457	0.598	0.591	0.604
	Prokaryote	40	**0.718**	0.697	0.668	0.659	0.658	0.673	0.659	0.631	0.712	0.703	0.669	0.532	0.665	0.595	0.647
	Virus	266	0.756	**0.788**	0.77	0.705	0.587	0.626	0.611	0.662	0.733	0.659	0.781	0.558	0.764	0.774	0.738
	Unknown	22	0.841	0.781	0.745	**0.85**	0.724	0.783	0.746	0.646	0.794	0.809	0.775	0.736	0.805	0.759	0.798
Page No.	2	386	0.734	**0.747**	0.734	0.684	0.587	0.63	0.614	0.634	0.719	0.662	0.744	0.551	0.732	0.727	0.712
	3	1	0.403	0.371	0.371	0.361	0.361	0.361	0.4	0.232	**0.642**	0.371	0.356	0.151	0.356	0.366	0.383
Pknot Type	H-type	330	0.734	**0.751**	0.735	0.688	0.581	0.62	0.602	0.637	0.719	0.649	0.742	0.545	0.731	0.731	0.709
	Kissing	22	0.634	0.57	0.626	0.573	0.573	0.58	**0.65**	0.444	0.553	0.642	0.638	0.619	0.586	0.554	0.621
	Complex	35	0.786	0.812	0.774	0.701	0.643	0.756	0.697	0.706	0.822	0.793	**0.824**	0.551	0.815	0.794	0.791
Winner Time(28)			2	6	0	2	1	0	1	0	**8**	2	7	0	0	0	0

Table D.4: The sensitivity of predicting missing set.

Name	Length	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfo ld5	pknots RG-M	pknots RG-F	pkn ots
3JYX_5	3170		**0.523**	0.479				0.475								
3KIY_A	2848		0.458	0.445				**0.674**								
2WDL_A	2807		0.443	0.42				**0.675**								
3J20_2	1495		0.58	0.617				**0.748**								
3ZEX_B	1465		0.364	**0.378**				0.276								
PKB192	1248		**0.529**	**0.529**				**0.529**			**0.529**			0.0	0.0	
3J2C_N	927		0.593	0.508				**0.75**			0.415			0.69	0.665	
PKB64	920		0.0	0.0	0.0			**0.625**			0.0			0.0	0.375	
PKB239	412	**0.556**	0.407	0.407	0.407	0.407	0.407	0.259		0.148	0.259		0.37	0.0	0.0	
3IZ4_A	377	0.221	0.589	0.589	0.6	0.6	0.6	**0.747**		0.516	0.463		0.347	0.579	0.653	
PKB149	351	0.219	0.156	0.156	0.75	0.75	0.75	**0.781**		0.688	0.219		0.438	0.75	0.5	
3IYQ_A	349	0.235	0.529	0.529	0.333	0.333	0.333	0.392		**0.686**	0.314		0.373	0.333	0.353	
PKB193	341	0.2	0.7	0.7	0.567	0.567	0.567	0.5		**0.767**	0.6		0.2	0.567	0.567	
PKB129	313	0.818	0.727	0.727	0.818	0.818	0.818	**0.909**		0.818	0.818		0.273	0.818	0.818	
PKB208	237	0.259	0.519	0.519	0.667	0.519	0.926	0.667		**1.0**	0.704		0.148	0.519	0.667	
PKB209	234	0.0	0.667	0.259	0.333	0.333	0.333	0.741		0.333	**0.889**		0.0	0.333	0.0	
PKB171	224	0.792	0.792	**1.0**	0.792	0.792	0.792	0.75		**1.0**	**1.0**		0.417	0.708	0.708	

Continued On Next Page

Table D.4 – *Continued From Previous Page*

Name	Length	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfo ld5	pknots RG-M	pknots RG-F	pkn ots
PKB77	219	0.35	0.35	0.35	0.7	0.7	0.7	0.7		**0.95**	0.6		0.35	0.7	0.7	
PKB150	212	0.278	0.278	0.278	0.278	0.278	0.278	0.5		0.0	0.5		0.0	0.222	0.5	**0.778**
PKB181	207	0.682	0.409	0.409	0.364	0.364	0.364	0.545		**0.909**	0.409		0.773	0.364	0.409	0.545
PKB354	190	0.4	0.52	0.52	0.52	0.76	0.52	0.52		0.52	0.28	0.28	0.28	0.52	0.52	**0.88**
PKB358	190	0.0	**0.542**	**0.542**	**0.542**	**0.542**	**0.542**	**0.542**		**0.542**	**0.542**	**0.542**	0.0	**0.542**	**0.542**	0.25
PKB324	181	0.534	0.793	0.793	0.81	0.81	**0.897**	0.793		0.793	0.707	0.845	0.328	0.741	0.741	0.862
PKB323	180	0.424	0.864	0.864	0.606	**0.909**	**0.909**	0.773		0.712	0.5	0.803	0.621	0.864	0.864	0.894
3ZEX_C	169	0.345	**0.448**	**0.448**	0.345	0.0	0.345	0.345		0.138	0.345	**0.448**	**0.448**	0.345	0.345	0.138
3PDR_A	161	0.8	**0.92**	**0.92**	0.8	0.8	0.8	0.9		0.72	0.8	**0.92**	0.64	0.9	0.86	0.64
PKB357	160	0.56	0.64	0.64	**0.8**	**0.8**	**0.8**	0.56		**0.8**	0.56	**0.8**	0.0	0.64	0.56	0.72
Average Sensitivity		0.404	0.531	0.52	0.552	0.583	0.615	0.618		0.634	0.521	**0.663**	0.316	0.506	0.516	0.634
Winner Time(28)		1	5	6	2	3	4	**10**		8	4	4	1	1	1	2

Table D.5: The PPV of predicting missing set.

Name	Length	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfo ld5	pknots RG-M	pknots RG-F	pkn ots
3JYX_5	3170		0.26	0.238				**0.28**								
3KIY_A	2848		0.417	0.389				**0.681**								
2WDL_A	2807		0.409	0.38				**0.702**								
3J20_2	1495		0.513	0.532				**0.657**								
3ZEX_B	1465		0.289	0.294				**0.348**								
PKB192	1248		0.021	0.021				0.02			**0.022**			0.0	0.0	
3J2C_N	927		0.507	0.443				**0.7**			0.368			0.584	0.561	
PKB64	920		0.0	0.0	0.0			**0.019**			0.0			0.0	0.01	
PKB239	412	**0.165**	0.102	0.096	0.087	0.087	0.087	0.096		0.031	0.07		0.085	0.0	0.0	
3IZ4_A	377	0.214	0.538	0.538	0.5	0.5	0.5	**0.747**		0.386	0.44		0.3	0.519	0.544	
PKB149	351	0.083	0.051	0.049	0.222	0.222	0.222	**0.278**		0.198	0.067		0.143	0.226	0.157	
3IYQ_A	349	0.132	0.27	0.27	0.15	0.15	0.15	0.196		**0.297**	0.138		0.174	0.153	0.158	
PKB193	341	0.064	**0.208**	**0.208**	0.167	0.167	0.167	0.17		0.202	0.184		0.061	0.165	0.165	
PKB129	313	**0.225**	0.174	0.167	0.176	0.176	0.176	0.198		0.171	0.188		0.071	0.178	0.178	
PKB208	237	0.111	0.175	0.175	0.237	0.192	0.301	0.237		**0.31**	0.237		0.056	0.179	0.237	
PKB209	234	0.0	0.237	0.1	0.118	0.118	0.118	0.256		0.102	**0.369**		0.0	0.118	0.0	
PKB171	224	0.306	0.264	**0.312**	0.25	0.25	0.257	0.261		0.304	0.32		0.164	0.239	0.239	

Continued On Next Page

Table D.5 – *Continued From Previous Page*

Name	Length	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfo ld5	pknots RG-M	pknots RG-F	pkn ots
PKB77	219	0.132	0.103	0.103	0.206	0.206	0.206	0.212		**0.253**	0.2		0.109	0.206	0.194	
PKB150	212	0.1	0.083	0.077	0.078	0.078	0.078	0.18		0.0	0.188		0.0	0.071	0.15	**0.209**
PKB181	207	0.242	0.134	0.134	0.118	0.118	0.118	0.214		0.247	0.145		**0.27**	0.119	0.138	0.169
PKB354	190	0.179	0.217	0.217	0.206	0.292	0.206	0.255		0.2	0.146	0.117	0.132	0.197	0.197	**0.344**
PKB358	190	0.0	0.228	0.228	0.232	0.232	0.22	**0.302**		0.22	0.228	0.236	0.0	0.22	0.22	0.097
PKB324	181	0.62	0.852	0.852	0.81	0.81	0.929	**0.979**		0.807	0.82	0.925	0.432	0.878	0.878	0.781
PKB323	180	0.5	0.877	0.877	0.667	0.938	0.938	0.944		0.81	0.524	0.898	0.732	**0.95**	**0.95**	0.843
3ZEX_C	169	0.233	0.277	0.277	0.189	0.0	0.189	**0.556**		0.071	0.286	0.232	0.289	0.208	0.182	0.069
3PDR_A	161	0.851	**0.92**	**0.92**	0.784	0.784	0.784	0.9		0.692	0.816	0.902	0.711	0.865	0.811	0.627
PKB357	160	0.326	0.327	0.327	0.392	0.392	0.392	0.35		0.364	0.378	**0.435**	0.0	0.34	0.269	0.34
Average PPV		0.236	0.313	0.305	0.279	0.301	0.318	0.398		0.298	0.279	**0.535**	0.196	0.292	0.284	0.387
Winner Time(28)		2	2	3	0	0	0	**12**		3	2	1	1	1	1	2

Table D.6: The MCC of predicting missing set.

Name	Length	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfo ld5	pknots RG-M	pknots RG-F	pkn ots
3JYX_5	3170		**0.368**	0.337				0.364								
3KIY_A	2848		0.436	0.416				**0.677**								
2WDL_A	2807		0.425	0.399				**0.688**								
3J20_2	1495		0.545	0.573				**0.7**								
3ZEX_B	1465		0.324	**0.333**				0.309								
PKB192	1248		0.104	0.104				0.102			**0.109**			-0.0	-0.0	
3J2C_N	927		0.548	0.474				**0.724**			0.39			0.635	0.610	
PKB64	920		-0.0	-0.0	-0.0			**0.108**			-0.0			-0.0	0.062	
PKB239	412	**0.302**	0.202	0.197	0.186	0.186	0.186	0.156		0.065	0.133		0.176	-0.002	-0.002	
3IZ4_A	377	0.215	0.562	0.562	0.546	0.546	0.546	**0.746**		0.444	0.449		0.32	0.546	0.594	
PKB149	351	0.133	0.087	0.085	0.407	0.407	0.407	**0.465**		0.368	0.119		0.248	0.411	0.278	
3IYQ_A	349	0.174	0.376	0.376	0.221	0.221	0.221	0.275		**0.449**	0.205		0.252	0.223	0.234	
PKB193	341	0.111	0.38	0.38	0.306	0.306	0.306	0.29		**0.392**	0.33		0.108	0.304	0.304	
PKB129	313	**0.428**	0.354	0.347	0.379	0.379	0.379	0.423		0.373	0.39		0.138	0.381	0.381	
PKB208	237	0.167	0.298	0.298	0.395	0.313	0.526	0.395		**0.556**	0.406		0.088	0.302	0.395	
PKB209	234	-0.004	0.395	0.158	0.195	0.195	0.195	0.434		0.181	**0.571**		-0.004	0.195	-0.005	
PKB171	224	0.491	0.455	0.557	0.443	0.443	0.449	0.44		0.55	**0.564**		0.259	0.41	0.41	

Continued On Next Page

Table D.6 – *Continued From Previous Page*

Name	Length	Cylo Fold	Dot Knot	DotK not-K	HotK nots-cc	HotK nots-dp	HotK nots-re	IP knot	MC-F old	McG enus	McQ Fold	pK iss	vsfo ld5	pknots RG-M	pknots RG-F	pkn ots
PKB77	219	0.212	0.187	0.187	0.377	0.377	0.377	0.383		**0.489**	0.344		0.193	0.377	0.367	
PKB150	212	0.163	0.149	0.142	0.144	0.144	0.144	0.297		-0.005	0.304		-0.004	0.122	0.271	**0.401**
PKB181	207	0.404	0.231	0.231	0.203	0.203	0.203	0.339		**0.472**	0.241		0.455	0.205	0.235	0.301
PKB354	190	0.264	0.332	0.332	0.324	0.469	0.324	0.361		0.319	0.198	0.176	0.188	0.316	0.316	**0.548**
PKB358	190	-0.005	0.348	0.348	0.351	0.351	0.342	**0.402**		0.342	0.348	0.355	-0.005	0.342	0.342	0.151
PKB324	181	0.572	0.82	0.82	0.808	0.808	**0.912**	0.88		0.798	0.759	0.883	0.371	0.805	0.805	0.819
PKB323	180	0.455	0.869	0.869	0.632	**0.922**	**0.922**	0.853		0.757	0.507	0.848	0.671	0.905	0.905	0.867
3ZEX_C	169	0.278	0.348	0.348	0.25	-0.007	0.25	**0.435**		0.092	0.31	0.318	0.356	0.263	0.245	0.091
3PDR_A	161	0.823	**0.919**	**0.919**	0.79	0.79	0.79	0.899		0.703	0.806	0.91	0.671	0.881	0.833	0.63
PKB357	160	0.423	0.454	0.454	0.557	0.557	0.557	0.439		0.536	0.457	**0.587**	-0.007	0.463	0.384	0.491
Average MCC		0.295	0.389	0.379	0.376	0.401	0.423	0.466		0.415	0.361	**0.582**	0.235	0.367	0.362	0.477
Winner Time(28)		2	2	2	0	1	2	**9**		5	3	1	0	0	0	2

Bibliography

Abrahams, J. P., M. van den Berg, E. Van Batenburg, and C. Pleij

1990. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucleic Acids Research*, 18(10):3035.

Akutsu, T.

2000. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104(1):45–62.

Andronescu, M., D. Dees, L. Slaybaugh, Y. Zhao, A. Condon, B. Cohen, and S. Skiena

2003. Algorithms for testing that sets of DNA words concatenate without secondary structure. *Natural Computing*, 2(4):391–415.

Baranov, P. V., R. F. Gesteland, and J. F. Atkins

2002. Recoding: translational bifurcations in gene expression. *Gene*, 286(2):187–201.

Baranov, P. V., O. L. Gurvich, O. Fayet, M. F. Prère, W. A. Miller, R. F. Gesteland, J. F. Atkins, and M. C. Giddings

2001. RECODE:a database of frameshifting, bypassing and codon redefinition utilized for gene expression. *Nucleic acids research*, 29(1):264–267.

Bekaert, M., L. Bidou, A. Denise, G. Duchateau-Nguyen, J.-P. Forest, C. Froidevaux, I. Hatin, J.-P. Rousset, and M. Termier

2003. Towards a computational model for -1 eukaryotic frameshifting sites. *Bioinformatics*, 19(3):327–335.

Bellaousov, S. and D. H. Mathews

2010. ProbKnot: fast prediction of RNA secondary structure including pseudoknots. *RNA*, 16(10):1870–1880.

Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne
2000. The Protein Data Bank. *Nucleic acids research*, 28(1):235–242.

Bindewald, E., T. Kluth, and B. A. Shapiro
2010. CyloFold: secondary structure prediction including pseudoknots. *Nucleic acids research*, 38(suppl 2):W368–W372.

Bon, M., C. Micheletti, and H. Orland
2012. McGenus: a Monte Carlo algorithm to predict RNA secondary structures with pseudoknots. *Nucleic acids research*, P. gks1204.

Bon, M. and H. Orland
2011. TT2NE: a novel algorithm to predict RNA secondary structures with pseudoknots. *Nucleic acids research*, P. gkr240.

Bon, M., G. Vernizzi, H. Orland, and A. Zee
2008. Topological classification of RNA structures. *Journal of molecular biology*, 379(4):900–911.

Bourdeau, V., G. Ferbeyre, M. Pageau, B. Paquin, and R. Cedergren
1999. The distribution of RNA motifs in natural sequences. *Nucleic acids research*, 27(22):4457–4467.

Brancotte, B., B. Yang, G. Blin, S. Cohen-Boulakia, A. Denise, and S. Hamel
2015. Rank aggregation with ties: Experiments and analysis. *Proceedings of the VLDB Endowment*, 8(10).

Brégeon, D., P. Amar, J. Azé, J.-P. Forest, A. Baudin-Baillieu, O. Namy, M. Termier, C. Zeng, J.-P. Rousset, A. Denise, and C. Froidevaux
. In silico selection and in vivo characterization of synthetic -1 ribosomal frameshift sequences. submitted.

Brierley, I.

1995. Ribosomal frameshifting on viral RNAs. *Journal of General Virology*, 76(8):1885–1892.

Brierley, I., R. C. Gilbert, and S. Pennell

2008. RNA pseudoknots and the regulation of protein synthesis. *Biochemical Society Transactions*, 36(4):684–689.

Brierley, I., A. J. Jenner, and S. C. Inglis

1992. Mutational analysis of the "slippery-sequence" component of a coronavirus ribosomal frameshifting signal. *Journal of molecular biology*, 227(2):463–479.

Brierley, I., S. Pennell, and R. J. Gilbert

2007. Viral RNA pseudoknots: versatile motifs in gene expression and replication. *Nature Reviews Microbiology*, 5(8):598–610.

Brunel, C., R. Marquet, P. Romby, and C. Ehresmann

2002. RNA loop–loop interactions as dynamic functional motifs. *Biochimie*, 84(9):925–944.

Cannone, J. J., S. Subramanian, M. N. Schnare, J. R. Collett, L. M. D'Souza, Y. Du, B. Feng, N. Lin, L. V. Madabusi, K. M. Müller, et al.

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(1):2.

Cao, S. and S.-J. Chen

2006. Predicting RNA pseudoknot folding thermodynamics. *Nucleic acids research*, 34(9):2634–2652.

Chen, R., J. Mintseris, J. Janin, and Z. Weng

2003. A protein–protein docking benchmark. *Proteins: Structure, Function, and Bioinformatics*, 52(1):88–91.

Chen, X., S.-M. He, D. Bu, F. Zhang, Z. Wang, R. Chen, and W. Gao

2008. FlexStem: improving predictions of RNA secondary structures with pseudoknots by reducing the search space. *Bioinformatics*, 24(18):1994–2001.

Clote, P., S. Dobrev, I. Dotu, E. Kranakis, D. Krizanc, and J. Urrutia

2012. On the page number of RNA secondary structures with pseudoknots. *Journal of mathematical biology*, 65(6-7):1337–1357.

Condon, A., B. Davy, B. Rastegari, S. Zhao, and F. Tarrant

2004. Classifying RNA pseudoknotted structures. *Theoretical Computer Science*, 320(1):35–50.

course, A. B.

2014. 2can support portal: Sequence formats.

Crick, F. et al.

1970. Central dogma of molecular biology. *Nature*, 227(5258):561–563.

Darty, K., A. Denise, and Y. Ponty

2009. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25(15):1974.

Dawson, W. K., K. Fujiwara, and G. Kawai

2007. Prediction of RNA pseudoknots using heuristic modeling with mapping and sequential folding. *PLoS One*, 2(9):e905.

De Rijk, P. and R. De Wachter

1997. RnaViz, a program for the visualisation of RNA secondary structure. *Nucleic Acids Research*, 25(22):4679–4684.

Ding, Y.

2006. Statistical and Bayesian approaches to RNA secondary structure prediction. *Rna*, 12(3):323–331.

Ding, Y., C. Y. Chan, and C. E. Lawrence

2005. RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. *RNA*, 11(8):1157–1166.

Ding, Y. and C. E. Lawrence

2003. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic acids research*, 31(24):7280–7301.

Dirks, R. M. and N. A. Pierce

2003. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of computational chemistry*, 24(13):1664–1677.

Do, C. B., D. A. Woods, and S. Batzoglou

2006. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–e98.

Eyrich, V. A., M. A. Martı-Renom, D. Przybylski, M. S. Madhusudhan, A. Fiser, F. Pazos, A. Valencia, A. Sali, and B. Rost

2001. EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics*, 17(12):1242–1243.

Farabaugh, P. J.

1996. Programmed translational frameshifting. *Microbiological reviews*, 60(1):103.

Fontana, W., D. A. Konings, P. F. Stadler, and P. Schuster

1993. Statistics of RNA secondary structures. *Biopolymers*, 33(9):1389–1404.

Forest, J.-P.

2005. *Modélisation et détection automatique de sites de décalage de cadre en -1 dans les génomes eucaryotes*. PhD thesis, Université Paris-Sud 11. Thèse soutenue le 30 juin.

Gan, H. H., S. Pasquali, and T. Schlick

2003. Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design. *Nucleic acids research*, 31(11):2926–2943.

Gardner, P. P. and R. Giegerich

2004. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC bioinformatics*, 5(1):140.

Gardner, P. P., A. Wilm, and S. Washietl

2005. A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic acids research*, 33(8):2433–2439.

Giedroc, D. P. and P. V. Cornish

2009. Frameshifting RNA pseudoknots: structure and mechanism. *Virus research*, 139(2):193–208.

Grüner, W., R. Giegerich, D. Strothmann, C. Reidys, J. Weber, I. L. Hofacker, P. F. Stadler, and P. Schuster

1996. Analysis of rna sequence structure maps by exhaustive enumeration i. neutral networks. *Monatshefte für Chemie/Chemical Monthly*, 127(4):355–374.

Gultyaev, A. P., F. Van Batenburg, and C. W. Pleij

1995. The computer simulation of RNA folding pathways using a genetic algorithm. *Journal of molecular biology*, 250(1):37–51.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten

2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Han, B., B. Dost, V. Bafna, and S. Zhang

2008. Structural alignment of pseudoknotted RNA. *Journal of Computational Biology*, 15(5):489–504.

Haslinger, C. and P. F. Stadler

1999. RNA structures with pseudo-knots: Graph-theoretical, combinatorial, and statistical properties. *Bulletin of mathematical biology*, 61(3):437–467.

Hendrix, D. K., S. E. Brenner, and S. R. Holbrook

2005. RNA structural motifs: building blocks of a modular biomolecule. *Quarterly reviews of biophysics*, 38(03):221–243.

Hochsmann, M., T. Toller, R. Giegerich, and S. Kurtz

2003. Local similarity in RNA secondary structures. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, Pp. 159–168. IEEE.

Hofacker, I. L.

2003. Vienna RNA secondary structure server. *Nucleic acids research*, 31(13):3429–3431.

Hofacker, I. L., W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster

1994. Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 125(2):167–188.

Huang, C.-H., C. L. Lu, and H.-T. Chiu

2005. A heuristic approach for detecting RNA H-type pseudoknots. *Bioinformatics*, 21(17):3501–3508.

Ieong, S., M.-Y. Kao, T.-W. Lam, W.-K. Sung, and S.-M. Yiu

2003. Predicting RNA secondary structures with arbitrary pseudoknots by maximizing the number of stacking pairs. *Journal of Computational biology*, 10(6):981–995.

Jabbari, H., A. Condon, A. Pop, C. Pop, and Y. Zhao

2007. HFold: RNA pseudoknotted secondary structure prediction using hierarchical folding. In *Algorithms in Bioinformatics*, Pp. 323–334. Springer.

Jabbari, H., A. Condon, and S. Zhao

2008. Novel and efficient RNA secondary structure prediction using hierarchical folding. *Journal of Computational Biology*, 15(2):139–163.

Jacobs, J. L., A. T. Belew, R. Rakauskaite, and J. D. Dinman

2007. Identification of functional, endogenous programmed -1 ribosomal frameshift signals in the genome of *Saccharomyces cerevisiae*. *Nucleic acids research*, 35(1):165–174.

Janssen, S. and R. Giegerich

2014. The RNA shapes studio. *Bioinformatics*, P. btu649.

Ji, Y., X. Xu, and G. D. Stormo

2004. A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics*, 20(10):1591–1602.

Jiang, M., P. J. Tejada, R. O. Lasisi, S. Cheng, and D. S. Fechser

2010. K-partite RNA secondary structures. *Journal of Computational Biology*, 17(7):915–925.

Kim, N., K. N. Fuhr, and T. Schlick

2013. Graph applications to RNA structure and function. In *Biophysics of RNA Folding*, Pp. 23–51. Springer.

Leontis, N. B. and E. Westhof

2001. Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(4):499–512.

Leontis, N. B. and E. Westhof

2002. The annotation of RNA motifs. *Comparative and functional genomics*, 3(6):518–524.

Lyngsø, R. B. and C. N. Pedersen

2000a. Pseudoknots in RNA secondary structures. In *Proceedings of the fourth annual international conference on Computational molecular biology*, Pp. 201–209. ACM.

Lyngsø, R. B. and C. N. Pedersen

2000b. RNA pseudoknot prediction in energy-based models. *Journal of computational biology*, 7(3-4):409–427.

Macke, T. J., D. J. Ecker, R. R. Gutell, D. Gautheret, D. A. Case, and R. Sampath

2001. RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic acids research*, 29(22):4724–4735.

Mathews, D. H.

2006. Revolutions in RNA secondary structure prediction. *Journal of molecular biology*, 359(3):526–532.

Mathews, D. H. and D. H. Turner

2006. Prediction of RNA secondary structure by free energy minimization. *Current opinion in structural biology*, 16(3):270–278.

Matthews, B. W.

1975. Comparison of the predicted and observed secondary structure of T4 phage

lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Mattick, J. S. and I. V. Makunin

2006. Non-coding RNA. *Human molecular genetics*, 15(suppl 1):R17–R29.

Mazauric, M.-H., P. Licznar, M.-F. Prère, I. Canal, and O. Fayet

2008. Apical loop-internal loop RNA pseudoknots: a new type of stimulator of -1 translational frameshifting in bacteria. *Journal of Biological Chemistry*, 283(29):20421–20432.

McCaskill, J. S.

1990. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119.

Metzler, D. and M. E. Nebel

2008. Predicting RNA secondary structures with pseudoknots by MCMC sampling. *Journal of mathematical biology*, 56(1-2):161–181.

Moon, S., Y. Byun, H.-J. Kim, S. Jeong, and K. Han

2004. Predicting genes expressed via -1 and +1 frameshifts. *Nucleic acids research*, 32(16):4884–4892.

Mullenweg, M., R. Boren, M. Jaquith, A. Ozz, and P. Westwood

2011. WordPress.

Namy, O., S. J. Moran, D. I. Stuart, R. J. Gilbert, and I. Brierley

2006. A mechanical explanation of RNA pseudoknot function in programmed ribosomal frameshifting. *Nature*, 441(7090):244–247.

Nebel, M. E. and F. Weinberg

2012. Algebraic and combinatorial properties of common RNA pseudoknot classes with applications. *Journal of Computational Biology*, 19(10):1134–1150.

Nussinov, R. and A. B. Jacobson

1980. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Sciences*, 77(11):6309–6313.

Parisien, M. and F. Major

2008. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452(7183):51–55.

Pearson, W. R. and D. J. Lipman

1988. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448.

Pleij, C.

1993. APPENDIX 2: RNA Pseudoknots. *Cold Spring Harbor Monograph Archive*, 24(0).

Ponty, Y., M. Termier, and A. Denise

2006. GenRGenS: software for generating random genomic sequences and structures. *Bioinformatics*, 22(12):1534–1535.

Puton, T., L. P. Kozlowski, K. M. Rother, and J. M. Bujnicki

2013. CompaRNA: a server for continuous benchmarking of automated methods for RNA secondary structure prediction. *Nucleic acids research*, 41(7):4307–4323.

Reeder, J. and R. Giegerich

2004. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC bioinformatics*, 5(1):104.

Reeder, J., P. Steffen, and R. Giegerich

2007. pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows. *Nucleic acids research*, 35(suppl 2):W320–W324.

Reidys, C. M., F. W. Huang, J. E. Andersen, R. C. Penner, P. F. Stadler, and M. E. Nebel

2011. Topology and prediction of RNA pseudoknots. *Bioinformatics*, 27(8):1076–1085.

Ren, J., B. Rastegari, A. Condon, and H. H. Hoos

2005. HotKnots: Heuristic prediction of RNA secondary structures including pseudoknots. *Rna*, 11(10):1494–1504.

Reuter, J. S. and D. H. Mathews

2010. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC bioinformatics*, 11(1):129.

Rietveld, K., R. Van Poelgeest, C. W. Pleij, J. Van Boom, and L. Bosch

1982. The tRNA-like structure at the 3' terminus of turnip yellow mosaic virus RNA. differences and similarities with canonical tRNA. *Nucleic acids research*, 10(6):1929–1946.

Rivas, E. and S. R. Eddy

1999. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of molecular biology*, 285(5):2053–2068.

Rødland, E. A.

2006. Pseudoknots in RNA secondary structures: representation, enumeration, and prevalence. *Journal of Computational Biology*, 13(6):1197–1213.

Ruan, J., G. D. Stormo, and W. Zhang

2004. An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. *Bioinformatics*, 20(1):58–66.

Sato, K., Y. Kato, M. Hamada, T. Akutsu, and K. Asai

2011. IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics*, 27(13):i85–i93.

Saule, C., M. Régnier, J.-M. Steyaert, and A. Denise

2011. Counting RNA pseudoknotted structures. *Journal of Computational Biology*, 18(10):1339–1351.

Schmitt, W. R. and M. S. Waterman

1994. Linear trees and RNA secondary structure. *Discrete Applied Mathematics*, 51(3):317–323.

Schroeder, S. J.

2009. Advances in RNA structure prediction from sequence: new tools for generating hypotheses about viral RNA structure-function relationships. *Journal of virology*, 83(13):6326–6334.

Seetin, M. G. and D. H. Mathews

2012. RNA structure prediction: an overview of methods. In *Bacterial Regulatory RNA*, Pp. 99–122. Springer.

Shapiro, B. A.

1988. An algorithm for comparing multiple RNA secondary structures. *Computer applications in the biosciences: CABIOS*, 4(3):387–393.

Sheikh, S., R. Backofen, and Y. Ponty

2012. Impact of the energy model on the complexity of RNA folding with pseudoknots. In *Combinatorial Pattern Matching*, Pp. 321–333. Springer.

Sperschneider, J. and A. Datta

2008. KnotSeeker: Heuristic pseudoknot detection in long RNA sequences. *RNA*, 14(4):630–640.

Sperschneider, J. and A. Datta

2010. DotKnot: pseudoknot prediction using the probability dot plot under a refined energy model. *Nucleic acids research*, 38(7):e103–e103.

Spirollari, J., J. T. Wang, K. Zhang, V. Bellofatto, Y. Park, and B. A. Shapiro

2009. Predicting consensus structures for RNA alignments via pseudo-energy minimization. *Bioinformatics and biology insights*, 3:51.

Stadler, P. F. and C. Haslinger

1997. RNA structures with pseudo-knots-graph-theoretical and combinatorial properties.

Staple, D. W. and S. E. Butcher

2005. Pseudoknots: RNA structures with diverse functions. *PLoS biology*, 3(6):e213.

Taufer, M., A. Licon, R. Araiza, D. Mireles, F. Van Batenburg, A. P. Gultyaev, and M.-Y. Leung

2009. Pseudobase++: an extension of pseudobase for easy searching, formatting and visualization of pseudoknots. *Nucleic acids research*, 37(suppl 1):D127–D135.

Ten Dam, E., K. Pleij, and D. Draper

1992. Structural and functional aspects of RNA pseudoknots. *Biochemistry*, 31(47):11665–11676.

Theis, C., S. Janssen, and R. Giegerich

2010. Prediction of RNA secondary structure including kissing hairpin motifs. In *Algorithms in Bioinformatics*, Pp. 52–64. Springer.

Theis, C., J. Reeder, and R. Giegerich

2008. KnotInFrame: prediction of -1 ribosomal frameshift events. *Nucleic acids research*, 36(18):6013–6020.

Thompson, J. D., F. Plewniak, and O. Poch

1999. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1):87–88.

Tinoco Jr, I. and C. Bustamante

1999. How RNA folds. *Journal of molecular biology*, 293(2):271–281.

Van Batenburg, F., A. P. Gultyaev, C. Pleij, J. Ng, and J. Oliehoek

2000. PseudoBase: A database with RNA pseudoknots. *Nucleic Acids Research*, 28(1):201–204.

Waugh, A., P. Gendron, R. Altman, J. W. Brown, D. Case, D. Gautheret, S. C. Harvey, N. Leontis, J. Westbrook, E. Westhof, et al.

2002. RNAML: a standard syntax for exchanging RNA information. *RNA*, 8(06):707–717.

Westhof, E. and P. Auffinger

2000. RNA tertiary structure. *Encyclopedia of analytical chemistry*.

Witwer, C., I. L. Hofacker, and P. F. Stadler

2004. Prediction of consensus RNA secondary structures including pseudoknots. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 1(2):66–77.

Wong, T. K., T. W. Lam, W.-K. Sung, B. W. Cheung, and S.-M. Yiu

2011. Structural alignment of RNA with complex pseudoknot structure. *Journal of Computational Biology*, 18(1):97–108.

wwPDB

2014. Protein data bank contents guide: Atomic coordinate entry format description.

Xayaphoummine, A., T. Bucher, and H. Isambert

2005. Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots. *Nucleic acids research*, 33(suppl 2):W605–W610.

Zhang, K. and D. Shasha

1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262.

Zuker, M.

2000. Calculating nucleic acid secondary structure. *Current opinion in structural biology*, 10(3):303–310.

Zuker, M.

2003. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13):3406–3415.

Zuker, M. and D. Sankoff

1984. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621.

Zuker, M. and P. Stiegler

1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–148.