IDENTIFICATION AND ANALYSIS OF NON-CODING RNAS IN LARGE SCALE GENOMIC DATA

By

Rujira Achawanantakun

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2014

ABSTRACT

IDENTIFICATION AND ANALYSIS OF NON-CODING RNAS IN LARGE SCALE GENOMIC DATA

By

Rujira Achawanantakun

The high-throughput sequencing technologies have created the opportunity of large-scale transcriptome analyses and intensified attention on the study of non-coding RNAs (ncRNAs). NcRNAs pay important roles in many cellular processes. For example, transfer RNAs and ribosomal RNAs are involved in protein translation process; micro RNAs regulate gene expression; long ncRNAs are found to associate with many human diseases ranging from autism to cancer. Many ncRNAs function through both their sequences and secondary structures. Thus, accurate secondary structure prediction provides important information to understand the tertiary structures and thus the functions of ncRNAs.

The state-of-the-art ncRNA identification tools are mainly based on two approaches. The first approach is a comparative structure analysis, which determines the consensus structure from homologous ncRNAs. Structure prediction is a costly process, because the size of a set of putative structures increases exponentially with the sequence length [1]. Thus it is not practical for very long ncRNAs such as lncRNAs. The accuracy of current structure prediction tools is still not satisfactory, especially on sequences containing pseudoknots. An alternative identification approach that has been increasingly popular is sequence based expression analysis, which relies on next generation sequencing (NGS) technologies for quantifying gene expression on a genome-wide scale. The specific expression patterns are used to identify the type of ncRNAs. This method therefore is limited to ncRNAs that have medium to high expression levels and have unique expression patterns that are different from other ncRNAs.

In this work, we address the challenges presented in ncRNA identification using different approaches. To be specific, we have proposed four tools, grammar-string based alignment, KnotShape, KnotStructure, and LncRNA-ID. Grammar-string is a novel ncRNA secondary structure representation that encodes an ncRNA's sequence and secondary structure in the parameter space of a context-free grammar and a full RNA grammar including pseudoknots. It simplifies a complicated structure alignment to a simple grammar string-based alignment. Also, grammar-string-based alignment incorporates both sequence and structure into multiple sequence alignment. Thus, we can then enhance the speed of alignment and achieve an accurate consensus structure. KnotShape and KnotStructure focus on reducing the size of the structure search space to enhance the speed of a structure prediction process. KnotShape predicts the best shape by grouping similar structures together and applying SVM classification to select the best representative shape. KnotStructure improve the performance of structure prediction by using grammar-string based-alignment and the predicted shape output by KnotShape. LncRNA-ID is specially designed for lncRNA identification. It incorporates balanced random forest learning to construct a classification model to distinguish lncRNA from protein-coding sequences. The major advantage is that it can maintain a good predictive performance under limited or imbalanced training data.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my adviser Dr. Yanni Sun. Her decision to admit me as her PhD student five years ago provided me the precious opportunity to study in Michigan State University and led me to the world of bioinformatics. During these five years under her guidance, I have made continuous progress in several aspects, including reading research papers, proposing research topics, developing methods, designing experiments, and writing papers. More importantly, I have gradually improved my ability to both independently and collaboratively conduct in-depth analysis into sophisticated research problems and use scientific methodologies to solve the challenging problems. She also gave me a lot of suggestions on how to effectively demonstrate our work to the audience, especially to people from other research areas. This ability is very important in that it will profoundly determine my capability to collaborate in a team of people from different backgrounds.

I also want to thank other committee members Dr. C. Titus Brown, Dr. Pang-Ning Tan, and Dr. James R. Cole. They gave a lot of useful suggestions during the course of my PhD program. I also thank my lab mates Yuan Zhang, Jikai Lei, Cheng Yuan, and Jiao Chen. During these years, we have productive discussion and cooperation on various research topics and I have obtained great help from them. I gratefully acknowledge other faculties and staffs of CSE department, especially Dr. Jin Chen, Dr. Richard J. Enbody, Dr. Joyce Chai, Dr. Li Xiao, Linda Moore, and Norma Teague. I also owe a lot of thanks to my friends in MSU especially Wenjuan Ma. All of them give me a lot of support and help during these years.

My final and most important acknowledgement must go to my family in Thailand and Dr. Alice and Ken Whiren, my host family in the US. They always give me persistent and determined love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PROCEDURES

Chapter 1

Introduction

1.1 Non-coding RNAs (ncRNAs)

It has been suggested that less than 2% of the human genome codes for proteins [2]. A recent study found only one-fifth of transcription across the human genome is associated with protein-coding genes [3]. This indicates that the majority of the transcriptom is non-coding RNAs [4], which are not translated into protein but function directly as RNAs [5, 6]. NcRNAs have been identified as potential regulatory molecules that play diverse and important roles in many biochemical processes. For instance, two typical house keeping ncRNAs, tRNA and rRNA, are key components for protein synthesis. MicroRNAs (miRNAs) play critical regulatory roles via interactions with specific target mRNAs in many organisms [7, 8].

The most recently discovered class of ncRNAs is long non-coding RNAs (lncRNAs), which are generally defined as non-coding transcripts of 200 nucleotides [9, 10, 11]. Increasing evidence has shown that lncRNAs play important and diverse biological functions. For example, lncRNAs ANRIL and HOTAIR bind to chromatin-remodeling complexes PRC1 and PRC2 to alter chromatin and transcription. GAS5 lncRNA acts as a decoy for the GR transcription factor and prevents GR from binding to DNA and transcriptional activation. MALAT1 RNA binds to SR proteins to regulate mRNA alternative splicing, whereas BACE-1AS RNA binds to the complementary BACE-1 mRNA to regulate BACE-1 translation [12]. As a result, the dysfunctions of lncRNAs are

associated with a wide range of diseases ranging from neurodegeneration to cancer [13].

1.2 Functional ncRNA identification

The functions of many types of ncRNA are determined not only by their sequences but also by their secondary structures. Thus, comparative ncRNA identification must exploit both sequence and structural conservations.

Many types of ncRNAs function through both sequences and secondary structures, which describe base pair interactions in ncRNA sequences. For example, the cloverleaf structure and the stem-loop structure are prominent features of tRNAs and pre-miRNAs, respectively. Thus, ncRNAs' structural annotation is an important component in their functional annotation.

Many computational methods have been used to determine the native structures of ncRNAs. A native structure is a structure that forms conformationally folding in native state before forming the tertiary structure.

In computational ncRNA prediction, secondary structural stability provides the characteristic signal for distinguishing real RNA sequences from non-functional transcripts [14]. Two fundamental approaches to structure prediction are *ab initio* and comparative folding. The *ab initio* method predicts a structure from a single sequence. The majority of them [15, 16, 17, 18, 19] search for putative structures with the minimum free energy (MFE) using an experimental number of derived energy parameters. However, the gap between the free energy of the stable native state and the less stable non-native structures is often small [20]. Thus, misfolded conformations can form with high probabilities [21].

The more accurate strategy is based on a homology detection using alignment methods [22]. A homologous alignment relies on conservation testimony of multiple sequences that have a common

ancestor. The similarity between sequences provides more information yielding to higher accuracy in structure prediction. Although there are promising progresses, finding the native secondary structure is still difficult. In particular, identifying the pseudoknot, an important structural motif in many types of ncRNAs, poses a great challenge for existing methods. Predicting the minimum free energy secondary structure that includes pseudoknots has been proven to be NP-hard [23].

Despite promising output by existing alignment tools, many existing secondary structure representations are highly complicated, incurring high computational cost during alignment. Even with various heuristics or pruning techniques to reduce the time complexity, ncRNA structural alignment is still more computationally intensive than pure sequence alignment and scale poorly with the number and length of input sequences. Therefore, it remains important to develop an efficient and accurate structure prediction method for ncRNAs.

1.3 Long non-coding RNA identification

Homologous alignment is an effective method to identify functional ncRNAs that exhibit similar functionality between species. However, the primary limitation of both sequence and structure based homologous alignment is that it might not yield a satisfactory result for sequences with low conservation like lncRNAs. LncRNAs are found to have low sequence similarity [24], but are functionally conserved [25, 26]. Although there exist functionally related structures of lncRNAs, predicting structures of lncRNAs is computational expensive because they are generally long with several hundred to thousand nucleotides. The size of the folding space of an ncRNA sequence increases exponentially with the sequence length [1]. Thus homology based functional identification is not practical for lncRNAs.

A numbers of studies have integrated high-throughput genomic technologies such as microar-

rays and next-generation sequencing (NGS) to identify functional ncRNAs in past decades [27]. These technologies have explored the ability of scientists to detect various types of ncRNAs including lncRNAs. Nevertheless, lncRNAs are generally expressed at a low level [28] and are less conserved [10], which have impeded their discovery and functional studies. Moreover, lncRNAs share similarities with protein coding transcripts, in which they resemble messenger RNAs (mRNAs) with respect to transcript length and splicing structure [29]. Many lncRNAs are found to have coding potential. For instance, H19, Xist, Mirg, Gtl2, and KcnqOT1 all have putative ORFs greater 100 amino acids, but have been characterized as functional ncRNAs [30]. These challenges pose great challenges to lncRNA identification. Therefore, accurately distinguishing long non-coding from protein coding transcripts is a critical first step towards comprehensive biogenesis assessment for the understanding of genetic information underneath.

To address the challenges in ncRNA identification, we have proposed three tools: grammar-string, KnotShape and KnotStructure, and LncRNA-ID. Grammar-string is a novel secondary structure representation based on a context free grammar. By encoding secondary structures in grammar strings, ncRNA structural alignment is projected to a simple sequence alignment. We can then predict the secondary structures of ncRNAs with less computational complexity. KnotShape and KnotStructure use shapes of secondary structures to optimize the search space of structure prediction. It classify similar structures to the same group based on a topology of each structure. The number of shapes is much smaller than the number of initial structures. With a small potential structure set, a multiple sequence alignment therefore becomes more practical and efficient especially for structures with pseudoknots. LncRNA-ID is designed to identify lncRNAs from protein-coding transcripts. It applies balanced random forest learning to construct a classification model. Thus, LncRNA-ID has a great competency to maintain steady performance with imbalanced training data, where the number of protein coding transcripts and lncRNAs are intrinsically

greatly different [10].

Chapter 2

Grammar string: a novel ncRNA secondary structure representation

2.1 Background

Comparative ncRNA identification, which searches for ncRNAs through evidence of evolutionary conservation, is the state-of-the-art methodology for ncRNA finding. Stochastic context-free grammar (SCFG) [31] provides a powerful way to encode both the sequence and structural conservations. A successful application of SCFG is ncRNA classification, which classifies query sequences into annotated ncRNA families such as tRNA, rRNA, riboswitch families. Other secondary structure modeling representations such as base pair probability matrices [32, 33, 34], tree profiles [35, 36], stem graphs [37] etc. have been used in RNA alignment, an important step in novel ncRNA detection. These alignment methods first infer the possible structures of each input sequence and then conduct structural alignment, whose accuracy and efficiently are highly dependent on structural representations. Despite promising output by existing alignment tools, many existing secondary structure representations are highly complicated, incurring high computational cost during alignment. Even with various heuristics or pruning techniques to reduce the time complexity, ncRNA structural alignment are still more computationally intensive than pure sequence alignment and scale poorly with the number and length of input sequences. Therefore, it remains

important to develop an efficient and accurate structural modeling and comparison method.

In this work, we design a novel secondary structure representation and show its application in consensus structure derivation through multiple ncRNA alignment. The two contributions are listed below. First, we design and implement grammar string, a novel ncRNA secondary structure representation. A grammar string is defined on a special alphabet constructed from a carefully chosen context free grammar (CFG). It encodes how this CFG generates an ncRNA sequence and its secondary structure. Compared to other secondary structure representations, grammar strings are simple and can take advantage of well-developed algorithms on sequences or strings. For example, grammar strings can convert ncRNA alignment into sequence alignment without losing any structural conservation, rendering highly efficient RNA alignment algorithm. In addition, supporting theories for sequence alignment such as score table design and Karlin-Altschul statistics [38] can be applied to grammar string alignment. Beyond alignment, grammar strings have potential for applications such as ncRNA sequence database indexing, ncRNA clustering, profile HMM-based ncRNA classification etc. It is worth mentioning that other string-based secondary structure representations [39, 40, 41] exit. However, those methods focus on deriving ncRNAs' similarities without resorting to alignment and thus cannot be directly applied for consensus structure derivation from homologous ncRNAs.

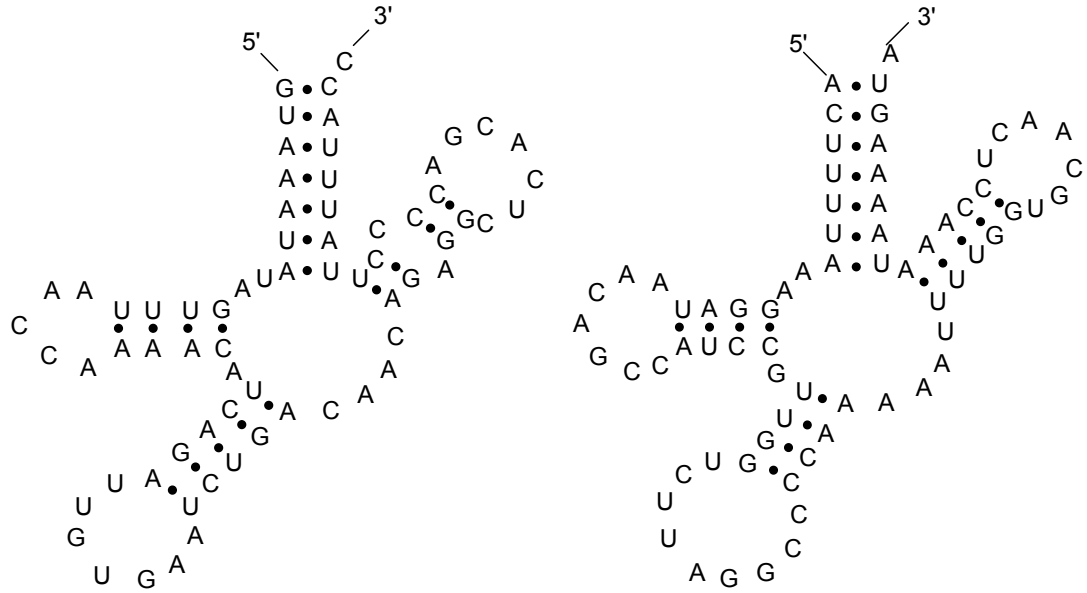
The second contribution is that we develop an effective method to exclude errors introduced by ab initio structure prediction. Many ncRNA alignment programs [35, 36, 32, 33, 34, 37] align predicted structured output by RNA folding tools. However, optimal prediction may not be the native structure [42], creating a need for choosing plausible structures as input to multiple alignment. In this work, we propose an efficient pattern matching method to pre-select predicted structures that are highly likely to be the true structure. This pre-screening can be used to reduce errors introduced by ab initio structure prediction and to remove contaminated sequences that are not homologous to

others.

Existing ncRNA alignment methods can be roughly classified into three basic types. The first type aligns and folds simultaneously. The most accurate algorithm of this type was developed by Sankoff [43]. However, it is prohibitively expensive with time complexity $O(L^{3N})$ and memory complexity $O(L^{2N})$, where L and N are the length and number of input sequences, respectively. Variants of the Sankoff algorithm have been proposed to reduce the computational time of multiple alignment, such as Stemloc [44], Consan [45], MARNA [46]. The second type of methods first builds a sequence alignment and then folds the alignment [47, 48, 22, 22, 49]. They infer structures from pre-aligned sequences generated using MULTIZ [50], ClustalW [51], or other available sequence alignment programs. The accuracy of these tools is largely affected by the alignment quality. In particular, when homologous ncRNA sequences only share structural similarity, building a meaningful sequence alignment becomes difficult. The third type of methods folds input sequences and then conducts structural alignment, yielding higher accuracy. Different tools in this category differ by different secondary structure modeling methods. Although some of them used restricted Sankoff algorithm in their implementations, we classify them into “fold and then align" category because they apply structure prediction in the first step. As our grammar string based alignment belongs to the third category, we discuss related “fold and then align" tools below, focusing on their secondary structure representations.

Several programs encode secondary structure using base pair probability matrices derived from McCastkill’s approach [18, 52]. NcRNA alignment is then converted into base pair probability matrix alignment. However, base pair probability matrix comparison is highly resource demanding. For example, pmcomp [32] takes $O(n^4)$ memory and $O(n^6)$ operations for aligning a pair of sequences with length n. More recent implementations such as LocARNA [33] and FOLDALIGNM [34] applied various restrictions or pruning techniques to reduce the time com-

plexity. But they are still much more expensive than sequence alignment.




```
tRNA_1: GUAAAUAUAGUUUAACCAAAACAUCAGAUUGUGAAUCUGACAACAGAGGCUCACGACCCCUUAUUUACC
        (((((((..((((.....)))).(((((.......)))))....((.(.......)).)))))))).
                          Grammar string
        cPPPPPPPUA#caac#aPPPPAACCA|PPPPPUUGUGAA|PPAcPPCUCACGA|

tRNA_2: ACUUUUAAAGGAUAACAGCCAUCCGUUGGUCUUAGGCCCCAAAAAUUUUGGUGCAACUCCAAAUAAAAGUA
        (((((((..((((.......)))).((((.........)))).....(((((.......)))))))))))).
                          Grammar string
        aPPPPPPPAA#uaaa#gPPPPAACAGCC|PPPPUCUUAGGCC|PPPPPUGCAACU|

                          Pairwise alignment
        cPPPPPPPUA#aac#aPPPPAAC--CA|PPPPP-UUGUGAA|PPPPPCUCACGA|
        aPPPPPPPAA#aaa#gPPPPAACAGCC|PPPPPCUUAGGC-|PPPPPUGCAACU|
         ******* **** * *******  * ****** **  *  ******  **   *
```

Figure 2.1 Two tRNA sequences from the human genome and the alignment of their grammar strings.The stars below the alignment denote exact matches.

RNAforester [35, 36] used tree profiles to represent secondary structures. Algorithms on tree alignment are applied for pairwise and multiple alignment computation. The asymptotic efficiency depends on the node number of the tree representation and the maximum degree d of a tree node. For n structures of average size s, their pairwise algorithm has time complexity $O(s^2d^2)$ and space complexity $O(s^2d)$. RNAforester can achieve higher efficiency than base pair probability matrix comparison. However, it is reported [33] that they tend to produce many alignment columns that

contain mostly gap characters in the multiple alignment mode. Carnac [37] used stem graphs to represent secondary structures. However, their program cannot accept more than 15 input sequences, limiting its practical usage.

2.2 Method

Inspired by Jaakkola and Haussler's discriminative classification method [53], we introduce **grammar string**, a representation of an ncRNA sequence in the parameter space of context-free grammar (CFG). Specifically, each ncRNA sequence and its secondary structure are transformed into a string defined on a new alphabet, where each character corresponds to a production rule in a CFG. We first introduce an unambiguous CFG for ncRNA sequence generation. Using the chosen CFG as an example, we formally define grammar strings for modeling an ncRNA sequence and its secondary structure.

2.2.1 An unambiguous CFG for ncRNA generation

NcRNA structures without pseudo-knots can be derived by CFGs [31]. A CFG is defined by a set of nonterminals, a set of terminals, a start nonterminal, and a set of production rules of the form $V \rightarrow \alpha$. V is a single nonterminal symbol, and α is a string of terminals and/or nonterminals. By recursively replacing nonterminals on the right hand side of each production rule, an ncRNA sequence and its secondary structure can be derived from a CFG. In this work, all our ncRNA sequences and their structures will be generated from G4, a light-weight CFG introduced by Dowell and Eddy [54], using leftmost derivation. Following the general definition of a CFG, G4 has a finite set of nonterminal symbols $V = \{\mathscr{S}, \mathscr{T}\}$, a finite set of terminal symbols $T = \{A, C, G, U, \varepsilon\}$, and a finite set of production rules defined as below:

- $\mathscr{S} \rightarrow a\mathscr{S}|\mathscr{T}|\varepsilon$
- $\mathscr{T} \rightarrow \mathscr{T}a|a\mathscr{S}\hat{a}|\mathscr{T}a\mathscr{S}\hat{a}$

where $a \in \{A,C,G,U\}$ and $\hat{a} \in \{A,C,G,U\}$. a and $\hat{a}$ form complementary base pairs such as A-U and G-C. In order to generate the unstructured single strand 'C' at 3' end and the two outmost base pairs in sequence tRNA_1 in Figure 2.1, the following production rules from G4 are called: $\mathscr{S} \rightarrow \mathscr{T}$, $\mathscr{T} \rightarrow \mathscr{T}$ C, $\mathscr{T} \rightarrow$ G $\mathscr{S}$ C, $\mathscr{S} \rightarrow \mathscr{T}$, $\mathscr{T} \rightarrow$ U $\mathscr{S}$ A. Continuing to replace $\mathscr{S}$ by correctly chosen production rules, we can derive tRNA_1. The sequence of production rules used for ncRNA structure generation is called a *derivation*.

Using the leftmost derivation, an *unambiguous* CFG can guarantee a *unique* derivation for a given ncRNA sequence and its secondary structure. For example, by using the unambiguous grammar G4, we have only one choice when choosing a production rule to derive tRNA_1 secondary structure in Figure 2.1. For a more detailed introduction about unambiguous CFGs, we refer readers to the review by Dowell and Eddy [54], where several light-weight unambiguous CFGs including G4 are discussed.

2.2.2 Grammar string generation algorithm

Each ncRNA secondary structure has a unique leftmost derivation from an unambiguous CFG, producing a one-to-one mapping between a structure and a production rule sequence. Intuitively, homologous ncRNAs with similar structures will share similar derivations. This motivates us to represent an ncRNA sequence and its secondary structure in the parameter space of a CFG. Thus, ncRNA structural comparison is converted to the comparison of their derivations.

In order to represent an ncRNA structure using its derivation, we introduce a new alphabet, where each character corresponds to a production rule in a CFG. One example alphabet derived

from G4 is defined below.

- Use upper case character of a to represent production rule $\mathscr{S} \rightarrow a\mathscr{S}$. For example, use A to represent $\mathscr{S} \rightarrow A\mathscr{S}$.
- Use | to represent $\mathscr{S} \rightarrow \varepsilon$.
- Use lower case character of a to represent production rule $\mathscr{T} \rightarrow \mathscr{T}a$. For example, use c to represent $\mathscr{T} \rightarrow \mathscr{T}C$.
- Use P to represent base pair emission $\mathscr{T} \rightarrow a\mathscr{S}\hat{a}$.
- Use a special character # to indicate branching $\mathscr{T} \rightarrow \mathscr{T}a\mathscr{S}\hat{a}$.
- No character is needed for production rule $\mathscr{S} \rightarrow \mathscr{T}$.

Thus, the new alphabet is $\mathscr{A} = \{$ A,C,G,U, a, c, g, u, P, |, #$\}$. If these production rules are used on DNA sequences, we can simply replace $U(u)$ with $T(t)$. For brevity, we name a string defined on the above alphabet a **grammar string**. As an example, the derivation for generating the unstructured single strand 'C' at 3' end and the two outmost base pairs in sequence *tRNA*_1 of Figure 2.1 is: $\mathscr{S} \rightarrow \mathscr{T}$, $\mathscr{T} \rightarrow \mathscr{T}$ C, $\mathscr{T} \rightarrow$ G $\mathscr{S}$ C, $\mathscr{S} \rightarrow \mathscr{T}$, $\mathscr{T} \rightarrow$ U $\mathscr{S}$ A. Thus, the corresponding grammar string is "cPP" using the alphabet $\mathscr{A}$. Note that we don't distinguish different base pairs (i.e. A-U, G-C, and G-U if allowed) in a grammar string. All base pairs are represented as 'P' in order to maximize the alignment score between homologous ncRNAs that share high structural similarity but low sequence similarity. Figure 2.1 shows the utility of grammar strings in detecting structural similarity between two tRNA sequences from the human genome. Because of low sequence similarity, BLAST [55] fails to align them. However, their structural similarity yields a meaningful global alignment between their corresponding grammar strings with 69% identity.

```
void parse(i, j)
{
if i >= j
        print '|';
        return;
else if X_i is a single stranded base
        print uppercase of X_i;
        i++;
        parse(i,j);
else if  X_j is a single stranded base
        print lowercase of  X_j;
        j--;
        parse(i,j);
else if  X_i and X_j form a base pair
        print 'P';
        i++ and j--;
        parse(i,j);
else
        print '#';
        k = the position that forms a base pair with X_j;
        parse(i,k-1);
        parse(k,j);
}
```

Figure 2.2 Algorithm for generating a grammar string for substring $X_{i..j}$.

In theory, our grammar string generation process consists of two steps. First, write the production rule sequence for an ncRNA sequence and its secondary structure. Second, transform the sequence of production rules into a grammar string according to the definition of grammar string alphabet. In practice, we use an efficient dynamic programming algorithm to design a grammar string for an ncRNA structure directly, skipping the step of parsing an ncRNA sequence using a CFG. The algorithm has time complexity $O(L^2)$, where L is the length of the ncRNA sequence.

Let X be an ncRNA sequence with its predicted or annotated secondary structure. i and j are indexes in X. X_i is the base at position i. Figure 2.2 sketches the dynamic programming algorithm generating a grammar string for substring $X_{i..j}$. In order to generate the complete grammar string for sequence X, one should call parse$(1, \mathrm{L})$.

2.2.3 Grammar pattern for encoding stem structures

The number of stems and their relationship largely define the basic “shape" of a secondary structure. For example, the cloverleaf structure of a tRNA sequence consists of four stems: acceptor stem, D stem, anticodon stem, and TΨCG stem. The precursor structure of a miRNA usually contains only one stem. According to the definition of grammar strings, three characters P,#, and | encode the number and relative positions of all stems in an ncRNA secondary structure. If we simply remove all single stranded regions (i.e. substrings only consisting of A,C,G,U, a, c, g, u) from a grammar string, we can use a simplified grammar string to represent the abstract stem structure for an ncRNA sequence. For brevity, we name a simplified grammar string a **g**rammar pattern, which is a string defined on a reduced alphabet $\{P, \#, |\}$. A grammar string can be converted into a grammar pattern in two steps: 1) remove all substrings representing single stranded regions, and 2) reduce every substring consisting of only Ps as a single P. Thus, the grammar pattern for sequence tRNA_1 in Figure 2.1 is $P\#\#P|P|P|$, where each P denotes a stem. There are four Ps, denoting four stems. The end of each stem is marked by |. Number of # defines the number of bifurcations.

Different distributions of the same number of stems can yield highly different secondary structures. Figure 2.3 shows how grammar patterns can account for different structures with the same number of stems. Note that all these grammar patterns are generated using G4 as the chosen CFG. If other unambiguous CFGs are used to generate grammar strings for the same structures, different sets of grammar patterns might be produced.

Ignoring all single stranded regions and length of each stem, grammar patterns only provide a coarse-grained description of ncRNA secondary structures. However, because of the high efficiency of pattern matching, grammar patterns can be used to speed up grammar string comparison. For example, we do not expect significant structural similarities between a tRNA and a miRNA

sequence. Instead of using Needleman-Wunsch [56] like alignment algorithm between their grammar strings, a constant time grammar pattern matching program can be applied as a filtration step. This filtration is particularly important when we aim to derive the consensus structure of multiple putatively homologous ncRNAs. Although these sequences are expected to be sequenced from the same gene family, it is possible that some of the sequences are from other regions. Thus, we can use the grammar pattern matching technique to exclude contaminated sequences, ensuring a multiple sequence alignment with good quality. The same technique can be used to remove possible errors introduced by MFE-based secondary structure prediction tools.

2.3 Using grammar strings for multiple ncRNA structural alignment

In this work, we show the utility of grammar strings in deriving consensus structure through multiple ncRNA alignment, which has wide applications in both known ncRNA classification and novel ncRNA search.

2.3.1 Score table design for grammar string alignment

Pairwise alignment is a fundamental step to multiple alignment and clustering. Existing alignment algorithms such as Needleman-Wunsch [56] can be directly applied to grammar strings when a score table defined on grammar strings' alphabet is imported. Following the common practice in score table design, we use maximum-likelihood ratio to derive the score between every pair of characters in grammar strings' alphabet $\mathscr{A}$. For each pair of characters a,b in $\mathscr{A}$, the score between a,b is $s(a,b) = \log \frac{\Pr(a,b)}{\Pr^0(a,b)}$. $\Pr(a,b)$ is the target probability of a,b in a set of true alignments

and $Pr^0(a,b)$ is the background probability that a and b are aligned. Assuming that a and b are independent in the background model, we get $\Pr^0(a,b) = Pr^0(a) \times Pr^0(b)$. Because the ncRNA family database Rfam [57] provides a large number of annotated ncRNA sequences, their alignments, and their associated secondary structures, we obtain both the target and the background probabilities from Rfam. In summary, we present following steps of designing a score table for grammar string alignment.

1. Build an alignment training set by randomly picking a large number of pairwise ncRNA alignment from Rfam 9.1's seed alignments. Some criteria are applied to select alignments with reasonably high quality. For example, if a pairwise alignment contains too many gaps, it will not be included in the training set. After applying the selection criteria, we had 18487 pairwise alignments in the training set.

2. Transform each pair of ncRNA sequence alignment into an alignment between grammar strings using the given secondary structure annotations by Rfam.

3. Compute the target probability $Pr(a,b)$ for each pair of aligned characters a,b in the above grammar string alignments.

4. Generate grammar strings for a large number of ncRNA sequences that are randomly picked from full families of Rfam 9.1. Compute the background probabilities $Pr^0(a)$ and $Pr^0(b)$ from these grammar strings.

The complete score table for grammar string alignment can be found at our website[1]. All exact matches have big positive scores. And bifurcation starting character # and stem ending character | can only be aligned with themselves or cause a gap. This is consistent with our intuition because it is not meaningful to align a bifurcation character with a base pair or a single stranded base.

[1] http://www.cse.msu.edu/∼yannisun/grammar-string

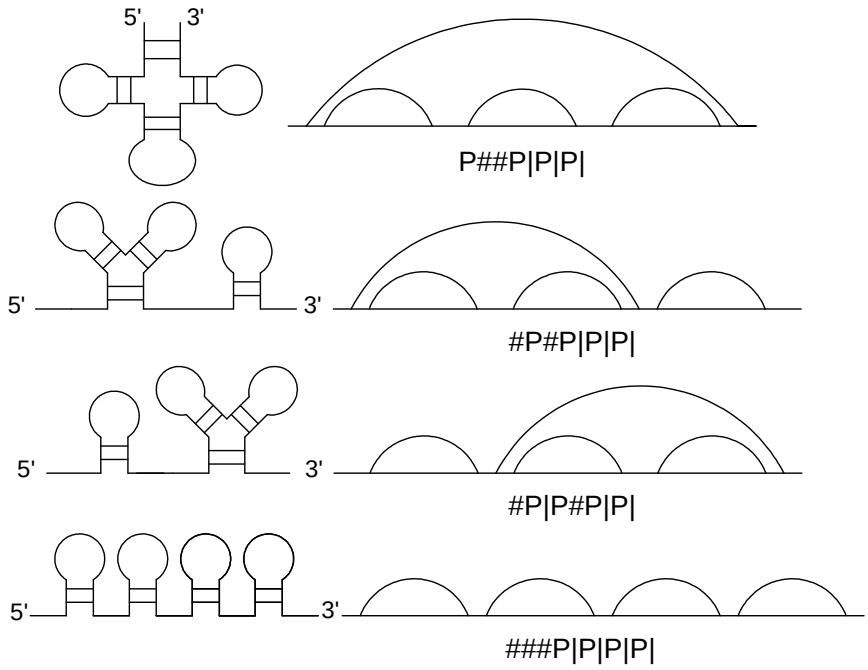


Figure 2.3 Four different stem structures and their grammar patterns. The left column shows the 2D representation of an ncRNA folding. The right column shows the distributions of stems along an ncRNA sequence. All grammar patterns are generated using G4 (our chosen unambiguous context-free grammar).

Insertions or deletions of 'P' or single stranded characters correspond to insertions or deletions of a base pair or single stranded bases in the ncRNA sequence alignment. Empirical experiments are conducted to choose default values for their gap opening and extension costs. The default gap opening score is slightly smaller than the lowest number in the grammar string's score table. The default gap extension cost is set as $1/10$ of the opening cost. We assign bigger gap penalties for structural characters # and $|$ in order to force corresponding stems or single stranded regions to be aligned together.

2.3.2 Multiple ncRNA alignment using grammar strings

Major steps of aligning multiple ncRNA sequences are sketched below.

1. Use an ab initio secondary structure prediction tool to predict both the optimal and sub-

optimal structures of each input sequence.

2. Generate a grammar string for each predicted secondary structure. If an ncRNA sequence has more than one structure predicted, multiple grammar strings will be generated.

3. Transform each grammar string into a grammar pattern. Use a voting mechanism to choose the most popular grammar pattern that mostly likely represents the native stem structure shared by the input sequences. All grammar strings that are not consistent with the chosen grammar pattern will be discarded.

4. Apply a progressive multiple sequence alignment method on remaining grammar strings.

5. Derive the consensus secondary structure from multiple grammar string alignment. Transform grammar string alignment into ncRNA sequence alignment using the ncRNA sequences and their predicted structures as references.

2.3.2.1 Structure prediction

Various tools exist to predict the secondary structures of a single input sequence. A majority of them search for structures with the minimum free energy (MFE) using a large number of experimentally derived energy parameters. The representative implementations include Mfold [15], RNAstructure [16, 17], McCaskill's base pairing probability computation [18], etc. MFE-based methods can also be combined with other probabilistic models such as conditional log-linear models (CLLMs) in ContraFold [58] for structure prediction. In our experiments, we choose MFE based tool UNAFold [59, 60] for structure prediction because of the following reasons. 1) It has a user-friendly interface for both web-site based and standalone tools. 2) It can generate both the optimal and suboptimal structures. It is shown that a suboptimal prediction rather than the optimal one could be the "correct" structure [42]. Thus, being able to output suboptimal structure increases

the chance of correct structure prediction for each input sequence. Empirically, we also tested other folding tools such as ContraFold on our test sequences. However, no clear advantage was observed.

2.3.2.2 Multiple grammar string alignment

We apply progressive alignment to multiple grammar strings. In the first stage, a guide tree is built based on all-against-all pairwise similarities and unweighted pair group method with arithmetic mean (UPGMA). In the second stage, the multiple sequence alignment is grown using the guide tree. Sum-of-pairs score is used to evaluate the similarity between a character and a column in an alignment or between two columns from two alignments. When we build the guide tree, several methods are used to convert an alignment score to sequence distance. The first distance definition comes from Feng and Doolittle [61]: $D = -\ln \frac{S_{real}(ij)-S_{rand}(ij)}{S_{iden}(ij)-S_{rand}(ij)}$, where S_{real} is the observed alignment score between sequences i and j. S_{iden} is the average of the two scores of the two sequences comparing with themselves. S_{rand} is the alignment score between two random sequences with the same length and composition as i and j. We applied shuffling to sequence i and j to obtain S_{rand}. Besides the Feng and Doolittle distance conversion method, we also evaluated several other simple distance definitions. The “Simple Distance" model defines $D = 1/(S_{read}(ij)/L)$, where L is the alignment length. “No-random FD" model defines $D = -\ln \frac{S_{real}(ij)}{S_{iden}(ij)}$. Our empirical experimental results show that both “Simple Distance" and “No-random FD" generate better alignment than more complicated Feng and Doolittle distance.

2.3.3 Using grammar patterns to reduce errors caused by ab initio structure prediction

The predicted structures for the same ncRNA sequence can differ significantly. It is important to align only structures that are likely to be consistent with the native structure of the homologous sequences. UNAFold [59] allows users to control the number of produced suboptimal structures by specifying a range of allowed thermodynamic energy values ΔG. Suboptimal structures can be highly different from the optimal structure for some ncRNA sequences. For example, tRNAs, which have functional cloverleaf structures, can be folded in different ways with reasonably small ΔGs. Figure 2.4 shows four different structures output by UNAFold for one tRNA sequence. Even worse, the true structure may not always be the optimal prediction with the minimum ΔG. Thus, it is not plausible to only keep the optimal prediction as correct structures may come from the sub-optimal predictions. In this section, a grammar pattern based screening approach is introduced to remove the contamination of wrong predictions before alignment.

As MFE-based *ab initio* structure prediction has limited accuracy, we increase structural prediction accuracy by using both the MFE and sub-optimal predictions. As a result, multiple grammar strings are derived for a single ncRNA. However, only one grammar string from each ncRNA should be used for alignment. We thus choose a set of grammar strings so that the sum of their pairwise similarity is maximized. Let m be the size of input ncRNA set S. For each ncRNA s_i, N_i structures and their associated thermodynamic energy values ΔGs are predicted for s_i. Each predicted structure is converted into a grammar string. Thus, the output of the *ab initio* structure prediction is a set of grammar strings and their associated ΔG: $\{(s_1^1, \Delta G_1^1), (s_1^2, \Delta G_1^2),..., (s_1^{N_1}, \Delta G_1^{N_1}), ..., (s_m^{N_m}, \Delta G_m^{N_m})\}$. s_i^j is the grammar string derived from the jth structure prediction for s_i. ΔG_i^j is the associated thermodynamic energy value. The goal is to choose a grammar string x_i for

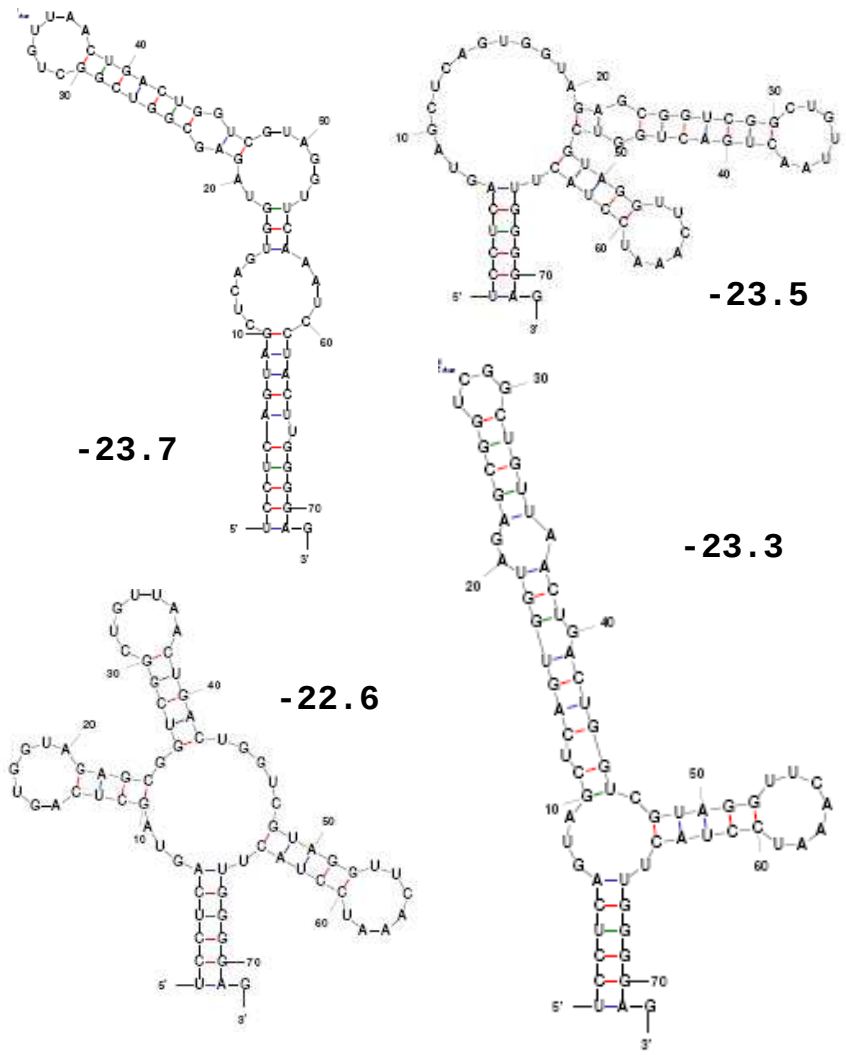


Figure 2.4 Four highly different structures predicted by UNAFold for the tRNA sequence shown at the bottom. The numbers beside each structure is their ΔG. The cloverleaf structure has a bigger ΔG than other predictions.

each ncRNA s_i so that the following function is satisfied:

$$argmax_{\{x_1,x_2,\ldots,x_m\}} \sum_{i,j \in S, i \neq j} sim(s_i^{x_i}, s_j^{x_j}) \tag{2.1}$$

where $sim(s_i^{x_i}, s_j^{x_j})$ is the similarity between two grammar strings derived from different ncRNAs.

$sim(s_i^{x_i}, s_j^{x_j})$ is minus infinity when $i == j$. Solving the above equation takes exponential running time. Thus, we propose two heuristics to reduce the time complexity. First, based on the observation that predicted structures for the same ncRNA can have highly different topologies (refer to Figure 2.4), we first determine the abstract shape using grammar pattern matching. Grammar strings that can be converted into the chosen grammar pattern constitute the *popular string set*. Others are discarded. Second, we apply an approximation algorithm on the popular string set to solve the above equation.

We first describe the algorithm of using the most popular grammar pattern as the representative abstract shape.

1. Denote the grammar pattern of grammar string s_i^x as o_i^x. Thus, for m ncRNAs, we have a set of grammar patterns and their associated free energy values ΔGs: $\{(o_1^1, \Delta G_1^1), (o_1^2, \Delta G_1^2),..., (o_1^{N_1}, \Delta G_1^{N_1}), ..., (o_m^{N_m}, \Delta G_m^{N_m})\}$. Usually $\Delta G < 0$.

2. Choose a grammar pattern that is shared by most input sequences. For each *different* grammar pattern o derived from the previous step, compute function:

$$f(o) = \sum_{i=1..m} \min_{x=1..N_i} \{\Delta G_i^x | o_i^x == o\} \tag{2.2}$$

   When the set $\{\Delta G_i^x | o_i^x == o\}$ is empty, $\min(\emptyset) = 0$. The grammar pattern o with the smallest $f(o)$ is the preferred structure of input ncRNA sequences. Denote this chosen grammar pattern as o^*.

3. Of multiple grammar strings generated for each ncRNA sequence, only grammar strings that can be converted to o^* are kept in the popular string set for further processing.

For m input sequences, there are $N_{total} = \sum_{1 \le i \le m} N_i$ structures predicted. Choosing the most pop-

ular grammar pattern has linear time complexity $O(N_{total})$.

Table 2.1 Structure predictions for three tRNA sequences. Multiple structure predictions are output for each sequence. For each prediction, column named "stems" displays its stem structure denoted by brackets. The corresponding grammar pattern and ΔG are listed in columns 3 and 4, respectively.

ID	stems	grammar pattern	ΔG
seq 1	(()()())	$P\#\#P\|P\|P\|$	-38.7 *
	(()()())	$P\#\#P\|P\|P\|$	-37
seq 2	(()()())	$P\#\#P\|P\|P\|$	-32.6 *
	(()()())	$P\#\#P\|P\|P\|$	-32
	(()()())	$P\#\#P\|P\|P\|$	-31.6
seq 3	()	P	-23.6
	(()())	$P\#P\|P\|$	-23

Once we have the popular string set, we choose m grammar strings from it to solve Eqn. (1). We apply an approximation algorithm based on guide tree building to choose x_i. Essentially, we build a guide tree from the popular string set until the tree contains a subtree with m leaves. As the similarity between grammar strings derived from the same ncRNA sequence is minus infinity, the first subtree with m leaves contains m grammar strings from m inputs with relatively small sum of all-pairs similarity. In order to build the tree, we need to conduct all-against-all comparison, which takes time complexity N_{total}^2. And, suppose there are d internal nodes in the tree when it finishes, the total running time is $O(N_{total}^2 + d\ N_{total})$.

An example of choosing grammar pattern and building popular string set for tRNAs is shown in Table 2.1. There are three different grammar patterns in Table 2.1: $P\#\#P|P|P|$, $P\#P|P|$, and P. Following the definition of $f(o)$ in Eqn. 2, f($P\#\#P|P|P|$) is the sum of ΔGs of grammar patterns denoted with *. Thus, f($P\#\#P|P|P|$) = -71.3, which is smaller than f(P) and f($P\#P|P|$). Therefore, $P\#\#P|P|P|$ is the chosen abstract shape for the three tRNAs. Note that no grammar string is chosen from "seq 3" because none of them can be converted to the chosen grammar pattern. Thus, the popular string set has size 5. we will build a guide tree from the five grammar strings which

are derived from “seq 1" and “seq 2". The guide tree stops at the first step because a subtree containing two leaves is formed. As a result, an alignment will only be conducted between two grammar strings with ΔG = -38.7 and -32.6.

Applying the same method to 20 tRNA sequences, we found the cloverleaf structure with four stems is the consensus structure shared by a majority of tRNA sequences. We repeated our experiments using different energy parameters. The dominant structure remains the cloverleaf structure although the second most popular structure alternates between a long hairpin and a three-stem structure (i.e. “(()())"). After discarding grammar strings that are not consistent with the chosen structure, we align remaining grammar strings using progressive alignment method.

2.4 Results

First, we conducted multiple sequence alignment for 20 tRNA sequences, which were used as an example of handling errors introduced by structure prediction programs in Section 2.3.3. Figure 2.5 shows the consensus secondary structure derived from aligning grammar strings of given tRNAs. We also tested other structural alignment programs including pmmulti [32], Murlet [62], RNAforester [35, 36], MARNA [46], and LocARNA [33]. Figure 2.5 shows that the grammar string alignment and Murlet both generate the best consensus structure for tRNA sequences.

Second, we use grammar strings to generate multiple ncRNA alignments for 452 families that are randomly chosen from BRAliBase 2.1, an enhanced RNA alignment benchmark [63]. This data set contains a diverse set of ncRNA families with different average sequence identity, length, and structural conservation. Each family contains 15 ncRNA sequences. Suboptimal structures with minimum-free energy values at most 10% higher than the optimal structure are predicted using UNAFold [59, 60] on over 6700 sequences from these 452 families. The average number of

Consensus grammar string using IUPAC code

aPPPPPPPUAu#cugn#rPPPPAGUUGGUA|PPPPPYUNANAA|PPPPPUUCRAAU|

Consensus sequence and secondary structure

XXXXXXXUAXXXXAGUUGGUAxxxxRXXXXXYUNANAAxxxxxNGUCXXXXXUUCRAAUxxxxxUxxxxxxxa
(((((((..((((........))))).(((((.......))))))....(((((.......))))).)))))).

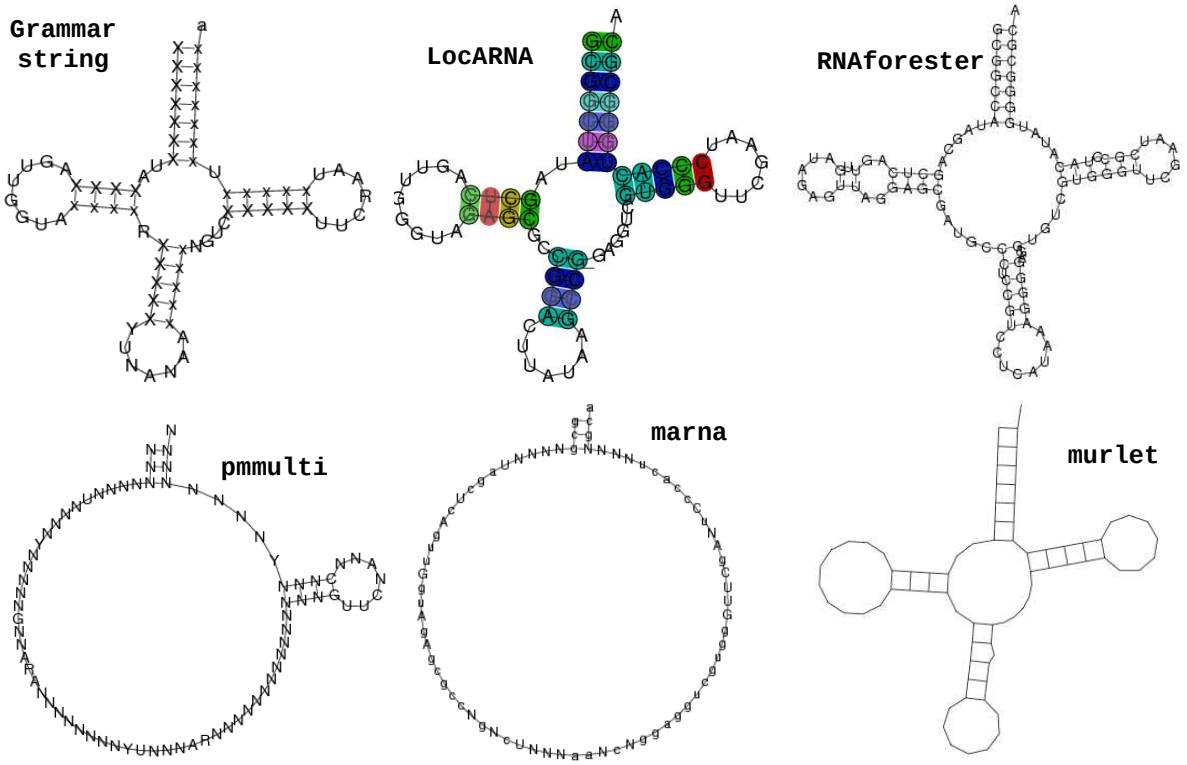


Figure 2.5 The consensus grammar string of tRNA alignment and the consensus secondary structure derived from the grammar string. X and x represent complementary base pairs. They can be easily translated into nucleotide bases using input tRNA sequences. All other structural alignment tools were tested under their default parameters except MARNA. For MARNA, using default structure prediction option RNAfold (from Vienna RNA package) generated no base pair in the consensus structure. Thus we used RNAsubopt, which yielded a few more base pairs in the consensus structure. The structure plotted by pmmmulti was generated from their consensus sequence and structure, which only included a very small number of base pairs. However, their multiple alignment seemed to contain more base pairs. RNAforester detected less number of complementary mutations and included several inconsistent base pairs such as U-U. LocARNA missed one base pair in one stem. Murlet generated the same structure as our grammar string alignment method.

suboptimal structure for each sequence is 20. For longer ncRNA sequences (length around 300), the number of suboptimal structures is close to 50. For short ones (length < 50), there are only a

couple of suboptimal structures predicted.

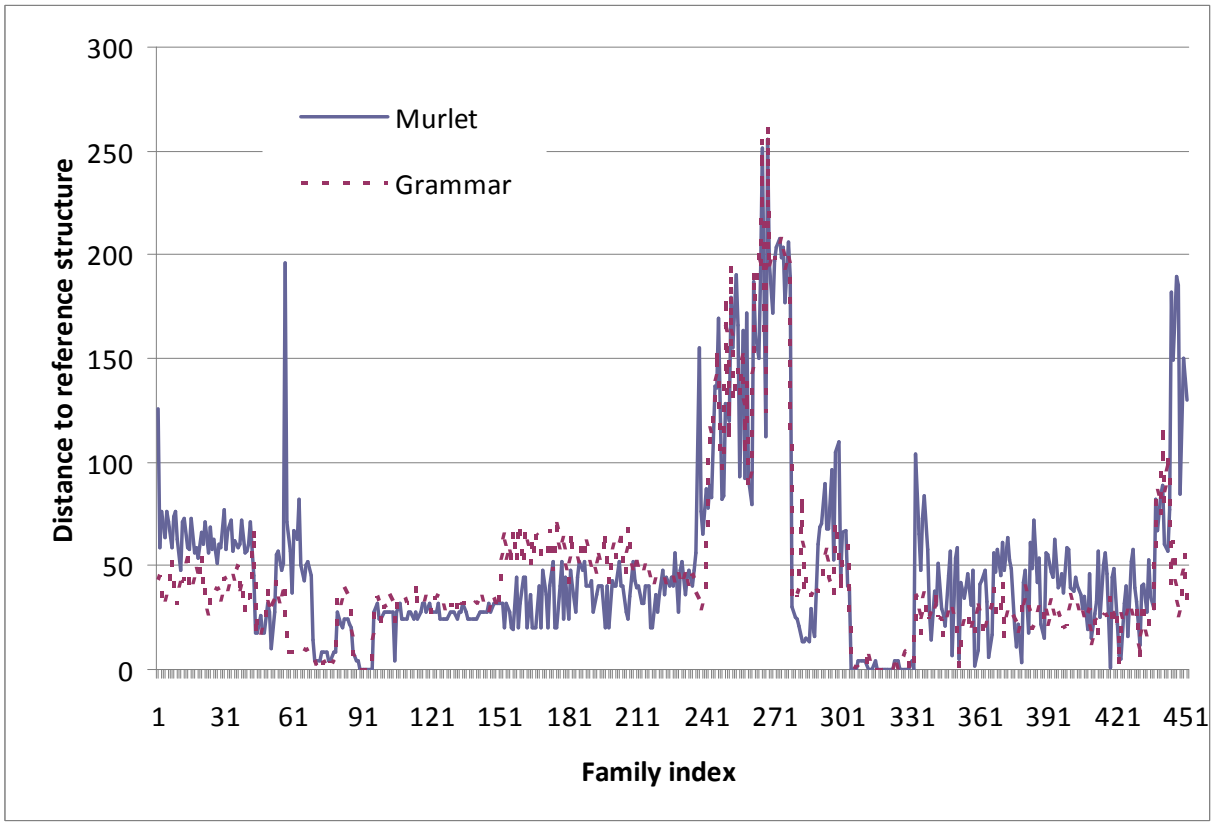


Figure 2.6 The differences of the reference structures (from Rfam) and the predicted consensus structures from grammar string and Murlet alignments are plotted and compared. Lower numbers indicate higher similarity between the predicted structure and the reference structure.

As Murlet [62], a Sankoff-based algorithm competes favorably in consensus structure quality with other ncRNA alignment tools, we compare the accuracy of consensus structures predicted from grammar string alignments and Murlet alignments. Since BRAliBase 2.1 only provides the alignments for each family of ncRNA sequences, but not their secondary structures, we extracted their reference structures from Rfam 9.1. In order to extract the consensus structure from a grammar string alignment, a consensus grammar string is first generated from the alignment (one example consensus grammar string is shown in Figure 2.5). And then this consensus grammar string is translated into a secondary structure using a reversed protocol to the one described in Figure 2.2. Murlet outputs the consensus structure along with each alignment. We compare the predicted sec-

ondary structures with the reference structures using RNAdistance from Vienna RNA package. Small distance indicates high similarity. The difference between predicted structures and the reference structures for both grammar string and Murlet alignments are summarized in Figure 2.6. Of 452 families, grammar string-based alignment produces consensus structures closer to the reference structures in 216 families and Murlet produces more accurate structure in 206 families. They generate the same consensus structures for 30 families. Some families pose hard cases for both methods, such as IRES_HCV and IRES_Picorna.

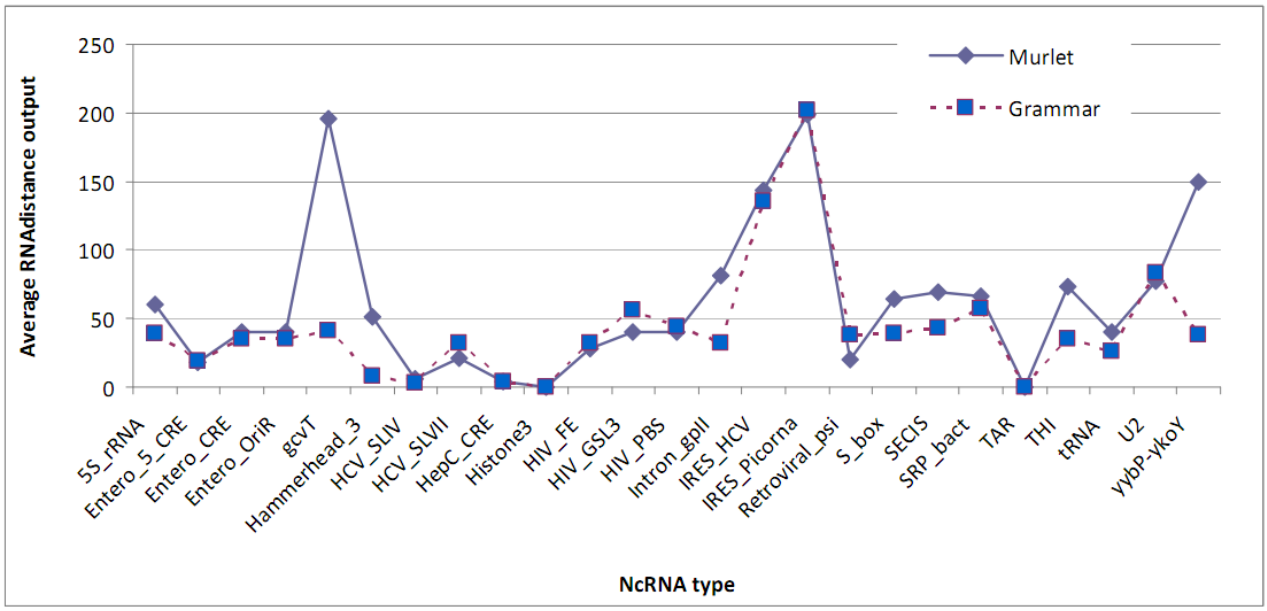


Figure 2.7 Consensus structures are derived for multiple families of each type of ncRNA, resulting a RNAdistance output vector. For each type of ncRNA, the average RNAdistance output for Murlet and grammar string alignment is compared.

Figure 2.8 compares the running time between Murlet and grammar string-based structure prediction. The running time of grammar string alignment largely depends on the size of popular string set, from which a set of strings are chosen as input to multiple alignment.

In order to analyze how grammar string and Murlet perform on each type of ncRNA, Figure 2.7 compares the average RNAdistance output for 25 types of ncRNAs, each of which contains multiple families in BRAliBase 2.1. The figure shows that grammar string-based methods produces

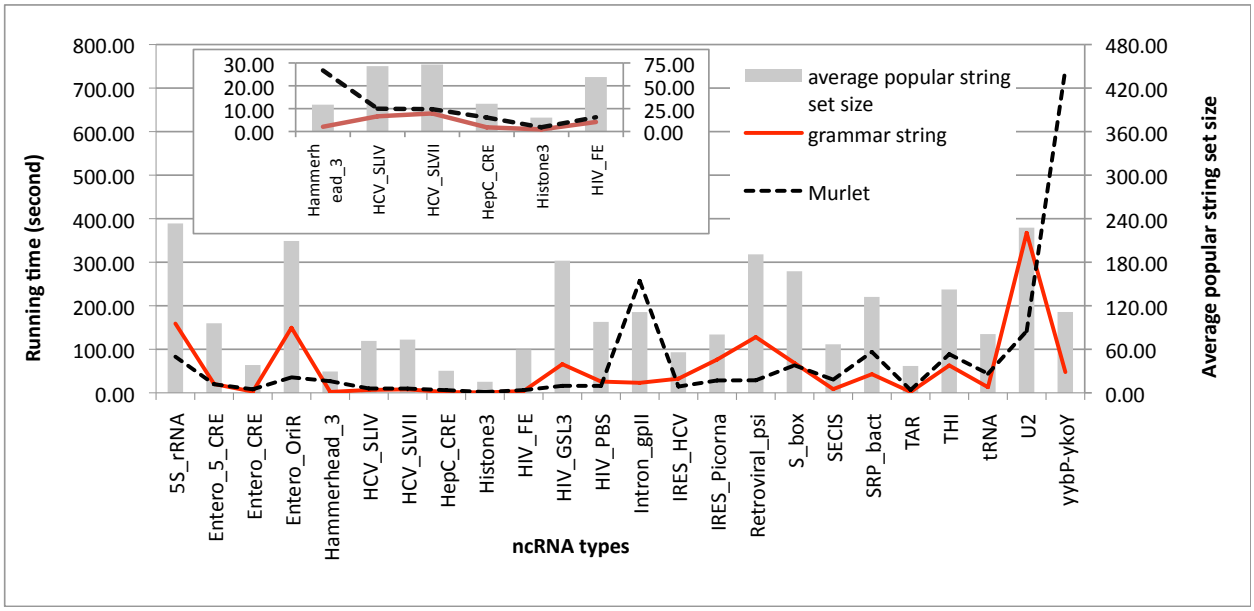


Figure 2.8 Running time comparison between grammar string and Murlet. The running time of grammar string is largely decided by the popular string set size. As Murlet uses over 2000 seconds for the family gcvT, we did not include this family in this figure in order to keep the fine scale of Y-axis.

more accurate consensus structures than Murlet for 13 types of ncRNAs: 5S_rRNA, Entero_OriR, gcvT, Hammerhead_3, HCV_SLIV, HepC_CRE, Intron_gpII, S_box, SECIS, SPR_bact, THI, tRNA, and yybP-ykoY. Murlet performs better for 10 types of ncRNAs: Entero_5_CRE, Entero_CRE, HCV_SLVII, HIV_FE, HIV|_GSL3, HIV_PBS, IRES_HCV, IRES_Picorna, Retroviral_psi, and U2. Thus, grammar string performs slightly better than Murlet in consensus structure derivation.

The major cause for the high structural difference for some families is the inaccuracy of the ab initio structure prediction program. Our alignment quality relies on the accuracy of structure prediction program. The prescreening algorithm can choose structures with the same number of stems and bifurcations. However, some predicted structures of homologous ncRNAs contain highly different numbers of base pairs for a pair of homologous sequences, causing low similarity between the derived grammar strings. Instead of using pure ab initio structure prediction tools,

we plan to use variants of Sankoff algorithm to generate consensus structures between a pair of sequences and then use these structures to derive grammar strings.

2.5 Discussion and conclusion

We have described the grammar string, a novel and simple ncRNA secondary structure representation. By encoding secondary structures in grammar strings, ncRNA structural alignment is transformed into sequence alignment. When there is no structural information available for ncRNA sequences, ab initio or other structure prediction tools are used to derive secondary structure information, which is needed for grammar string generation. Thus, grammar string alignment quality relies on the accuracy of structure prediction. When the structure prediction is reasonably accurate, grammar string alignment can be highly accurate and efficient for homologous ncRNA consensus structure derivation. Besides building ncRNA structural alignment, grammar string can be used to encode characterized ncRNA structures, comparing different structures, and searching for common structural motifs.

In the current grammar string generation algorithm, we don't distinguish different base pairs (G-C, A-U, and U-G if allowed) in order to maximize alignment score of homologous ncRNA sequences that share strong structural similarity rather than sequence similarity. However, it is worth testing whether an expanded alphabet can increase alignment accuracy. Thus we plan to : 1) distinguish different base pairs in an expanded grammar string alphabet, and 2) use a set of high quality pairwise ncRNA alignments to train the new substitution score table for the new alphabet. In addition, we will evaluate how different alignment methods (such as interactive vs. progressive) and different gap penalties in stems and single stranded regions affect the final alignment quality.

Chapter 3

Secondary structure prediction of ncRNAs including pseudoknots

3.1 Background

As knowing the secondary structure provides important information to understanding the tertiary structures and thus the functions of ncRNAs, deriving the secondary structures of ncRNAs remains an important research topic in RNA informatics. Pseudoknot is an important structural motif in secondary structures of many types of ncRNAs. Formally, a pseudoknot occurs when an RNA has two base pairs, $i-j$ and $i'-j'$, such that $i < i' < j < j'$. Psuedoknots are known to play important functions in telomerase RNA, tmRNA, rRNA, some riboswitch, some protein-biding RNA, Viral ribosomal frameshifting signals, etc [64].

There are two types of structure prediction methods. One is *ab initio* folding tools. A majority of them [15, 16, 17, 18, 19] search for structures with the minimum free energy (MFE) using a large number of experimentally derived energy parameters. Despite promising progress in the *ab initio* structure prediction methods, their accuracy is still limited. The predicted structure with the MFE may not be the native structure of an ncRNA.

More accurate structure prediction methods are based on comparative ncRNA analysis, which aligns homologous ncRNAs and derives their consensus structure. A number of ncRNA alignment

and structure derivation tools exist. The most accurate tool developed by Sankoff [43] aligns and folds ncRNAs simultaneously. However, this method is prohibitively expensive. Even with various heuristics or pruning techniques, ncRNA structural alignment tools are still computationally intensive and scale poorly with the number and length of input sequences. In addition, most existing tools do not allow pseudoknots, an important structural motif in RNAs. Thus, there is a need for an efficient and accurate comparative structure derivation tool that can handle any secondary structure including pseudoknots.

In this work, we introduce a consensus structure derivation approach based on **g**rammar string, a novel ncRNA secondary structure representation that encodes an ncRNA's sequence and secondary structure in the parameter space of a context-free grammar (CFG) and a full RNA grammar including pseudoknots. The main components of our method include:

- A novel ncRNA secondary structure representation named **g**rammar string, which is defined on a special alphabet constructed from production rules in a formal grammar such as context free grammar (CFG) [31]. It encodes how this grammar generates an ncRNA sequence and its secondary structure. Grammar strings are simple and can take advantage of well-developed algorithms on sequences or strings.

- A systematic method to exclude errors introduced by *ab initio* structure prediction. As the optimal structure output by existing tools may not be the native structure, we design an algorithm to choose correct structures using both the optimal and sub-optimal predictions.

- A general method that can be applied to any structural alignment as long as there is a formal grammar describing the structure. We have extended grammar strings to handle pseudoknots using the full RNA grammar introduced by Rivas and Eddy [65].

3.2 Method

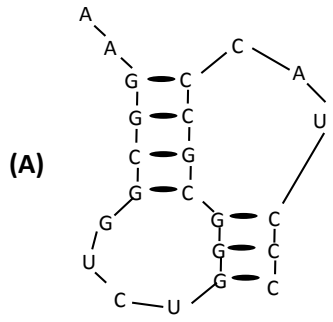
3.2.1 Grammar string and grammar pattern

Inspired by Jaakkola and Haussler's discriminative classification method [53], we introduce grammar string, a representation of an ncRNA sequence and its structure in the parameter space of a grammar. In this method, each ncRNA sequence and its secondary structure are transformed into a string defined on a new alphabet, where each character corresponds to a production rule in a grammar. We use two grammars in this work. For secondary structures without pseudoknots, a context-free grammar (CFG) is sufficient. For secondary structures containing pseudoknots, we choose the full ncRNA grammar introduced by Rivas and Eddy [65].

3.2.2 Grammar strings for ncRNAs with pseudoknots

Pseudoknot is an important structural motif for ncRNAs. Two helical segments are either parallel or nested for pseudoknot-free structures. But they can form crossover pseudoknot as shown in Figure 3.1.(A). Describing the language of pseudoknots is beyond the ability of CFG. Recently, Rivas and Eddy [65] presented a full RNA grammar for ncRNAs with pseudoknots. This grammar adopted a small number of auxiliary symbols instead of general context-sensitive grammar to parse sequences with pseudoknots in polynomial time. In this work, to distinguish it from G4 in the previous chapter, we name the full grammar "RE-pseudo".

RE-pseudo has a bigger set of nonterminal symbols V={$\mathscr{W}$, $\mathscr{W}_B$, $\mathscr{V}^{ab}$, $\mathscr{W}_H$, $\mathscr{V}_H^{abcd}$, IS_1, IS_2 } than G4, where $\mathscr{W}_H$ and $\mathscr{V}_H^{abcd}$ are used to generate pseudoknots. There are 43 different production rules in this grammar and thus we define grammar strings on an alphabet of size 43. The full list of production rules is not given here due to the space limitation. We refer readers to the original paper by Rivas and Eddy. We only list three production rules which will be used to generate the prefix

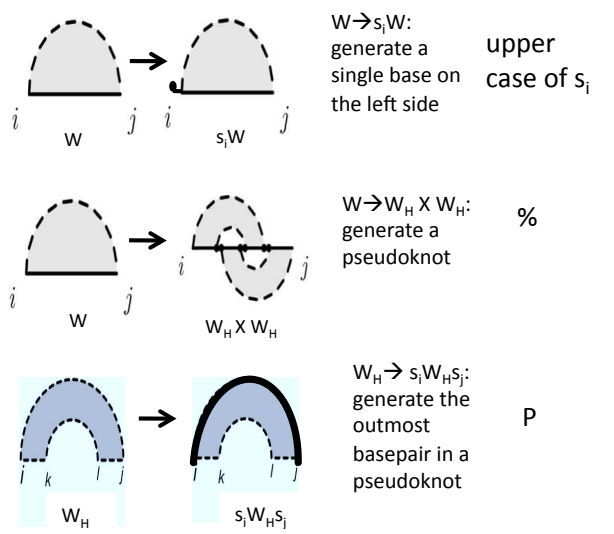


(A)

```
AAGGCGGUCUGGGCGCCCAUCCC
..<<<<....AAA>>>>...aaa
```

(B)

(C)



(D)

Applied rule	Char -acter	comment
$W \rightarrow s_1 W$	A	A on the 5'
$W \rightarrow s_2 W$	A	A on the 5'
$W \rightarrow W_H X W'_H$	%	Pseudoknot. W_H will generate GGCGGUCU…CGCC <<<< >>>> W'_H will generate remaining bases.
$W_H \rightarrow s_3 W_H s_{17}$	P	The first base pair GC inside W_H
$W_H \rightarrow s_4 W_H s_{16}$	P	The second base pair GC inside W_H

Figure 3.1 (A) An example of pseudoknot with the most common topology. (B) The dot-bracket representation for the pesudoknot in (A). <> and *Aa* are used to distinguish base pairs from two helical segments in the pseudoknot. (C) Three production rules from RE-pseudo. The diagrams are reproduced from Rivas and Eddy's original description of this grammar. (D) The first five steps in (A)'s derivation using the grammar RE-pseudo.

of a grammar string for the simple pseudoknot in Figure 3.1.(A). The three rules, their diagrams, meanings, and the assigned characters can be found in Figure 3.1.(C). Based on these conventions, the prefix for the H-ytpe pseudoknot in Figure 3.1.(A) is AA%PP. We can also replace P with different characters to distinguish different base pairs.

In the previous chapter, we use the pseudoknot-free grammar G4 as an example to introduce grammar pattern. The same idea can be applied to RE-pseudo. We can transform grammar strings for RE-pseudo into grammar patterns using a similar strategy. Although this grammar can replace

G4 for any ncRNA secondary structure derivation, it is much more complicated than G4. Thus, we keep G4 for ncRNAs without pseudoknots. The pseudocode for generating grammar string based on RE-pseudo parsing can be found on $http://www.cse.msu.edu/\ achawana/grammar-string$.

3.2.3 Consensus structure derivation through multiple grammar string alignment

In this section, we sketch the major steps of consensus structure derivation using grammar string alignment.

1. Choose an *ab initio* secondary structure prediction tool to predict both the optimal and sub-optimal structures for each input sequence.

2. Generate a grammar string for each predicted secondary structure. If an ncRNA sequence has more than one structure predicted, multiple grammar strings will be generated.

3. Based on the assumption that the correct structure should be shared by a majority of input sequences, we choose one grammar string from each input ncRNA so that the sum of their pairwise similarity is maximized. This step is used to remove possibly wrong predictions from previous *ab initio* structure prediction. Detailed algorithms can be found in the next section.

4. Apply progressive multiple sequence alignment on chosen grammar strings from the previous step and derive the consensus secondary structure.

Various tools [15, 16, 17, 18] exist to predict the secondary structures of a single input sequence. A majority of them search for structures with the minimum free energy (MFE) using a large number of experimentally derived energy parameters. For structure prediction without pseudoknots, we

choose UNAFold [59, 60] because: 1) it has been tested extensively; 2) it is easy to apply on large-scale experiments; 3) it can generate both the optimal and suboptimal structures. Empirically, we also tested other folding tools such as ContraFold [58] on our test sequences. However, no clear advantage was observed. For similar reasons, we choose Hotknots 2.0 [19] for pseudoknot prediction.

To derive the consensus secondary structure we conduct progressive multiple alignment on grammar strings using a guide tree, which is built based on all-against-all pairwise similarities and unweighted pair group method with arithmetic mean (UPGMA).

3.3 Results

Table 3.1 Comparison of grammar string and RNASampler on pseudoknot derivation

RfamID	name	average length	grammar string accuracy	RNA-Sampler accuracy	grammar string running time (s)	RNA-Sampler running time (s)
RF00165	Corona_pk3	62	4	8	2.96	2.59
RF00381	Antizyme_FSE	57	0	11	3.07	5.2
RF00505	RydC	64	0	3	2.78	0.32
RF00176	Tombus_3_IV	91	30		1.46	12.45
RF00233	Tymo_tRNA-like	82	11		1.21	6.7
RF00499	Parecho_CRE	111	29		2.22	2.77
RF00507	Corona_FSE	82	20		1.34	5.56
RF00523	Prion_pknot	41	4		0.51	0.82

Accuracy is defined as the base pair difference between the reference structure and the predicted structure. RNASampler outputs errors or structures that do not have the same abstract shape as the reference structure for five families. Thus their accuracy is not measured.

The current version of Rfam contains 71 families with pseudoknots. 25 of them are published rather than being computationally predicted. We focus on testing grammar strings on the 25 families. For each family, we first apply HotKnots [19] to predict secondary structures. Both the optimal and sub-optimal predictions are kept. As these predictions show great variance in their

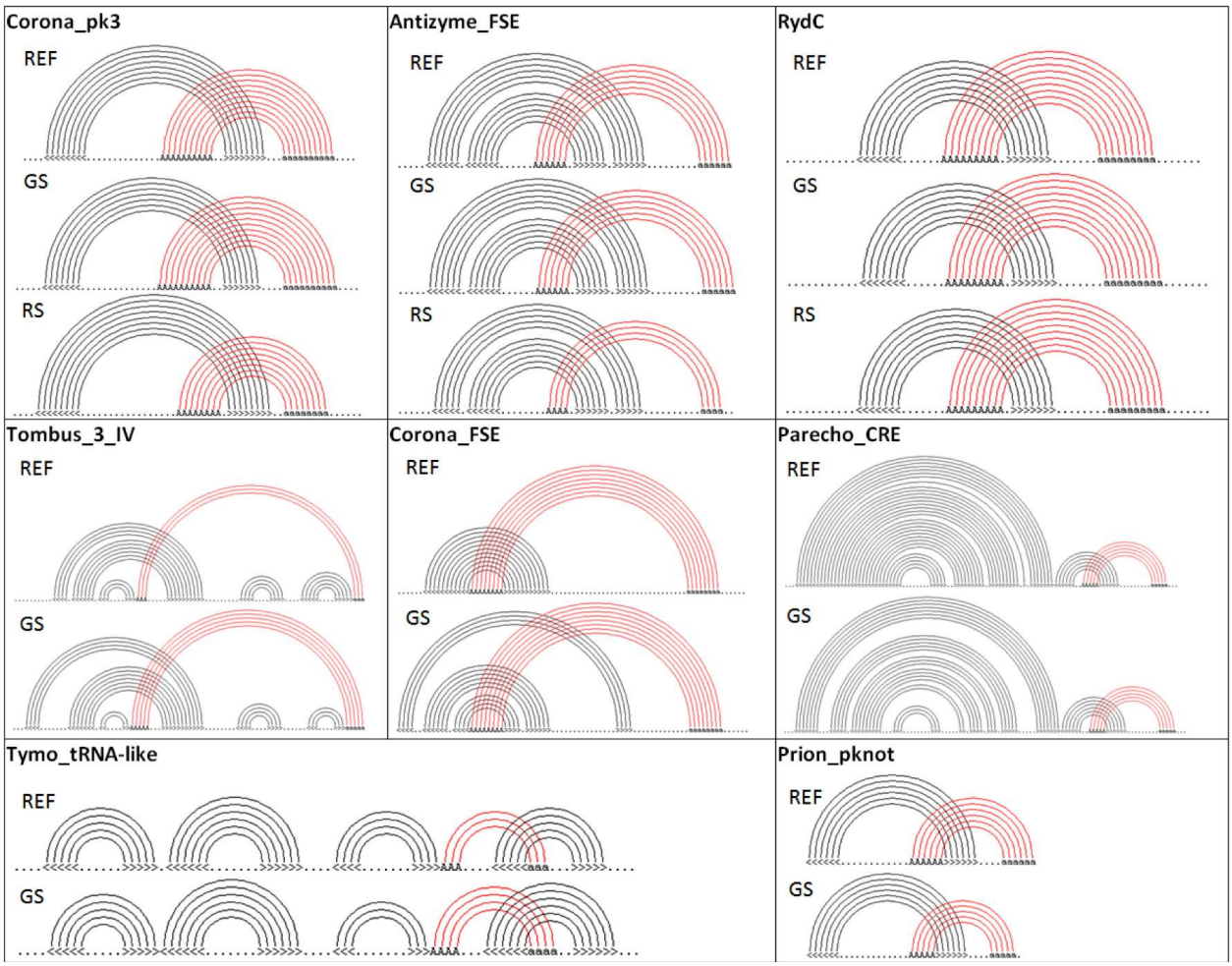


Figure 3.2 Illustration of the reference structures and predicted structures by grammar string-based alignment (denoted as GS) and RNASampler (denoted as RS) on 8 families. Note that if RNASampler fails to output a structure containing pseudoknots or outputs errors, their predictions are not displayed.

topology, we use the method described in Section 2.3.3 to choose a grammar pattern that is shared by a majority of input sequences. Only grammar strings that can be converted into the chosen grammar pattern are candidates for multiple alignment. For 17 families, we fail to obtain a grammar pattern that is shared by at least 2 input sequences. Thus, we can only conduct multiple ncRNA alignment for 8 families. For each family, if the number of available grammar strings is less than 10, we align all of them. Otherwise, at most 10 grammar strings are aligned. To compare our tool with existing ones that do not require an alignment as input, we apply RNASampler [66] on the

25 families. RNASampler outputs errors or structures without pseudoknots for 14 families. And for other three families, the predicted structures are not as accurate as grammar string alignment. ILM [67] can also predict pseudoknots without requiring an alignment as input. However, we are not able to run their program on our input. We summarize properties of the eight families, the structure derivation results of grammar string alignment and RNASampler in Table 3.1. We also plot the reference structures and the predicted structures from grammar strings and RNASampler in Figure 3.2. The results show that pseudoknot prediction is a highly challenging problem, especially for long ncRNAs ($>$ 100 nt). Hotknots becomes less accurate with the increase of inputs' sizes. Both our program and RNASampler are not able to handle long ncRNAs with pseudoknots. The commands, parameters, and alignment results of RNASampler and our tool can be downloaded from our website.

3.4 Discussion and conclusion

By encoding secondary structures in grammar strings, ncRNA structural alignment is transformed into sequence alignment. We have shown the utility of grammar string alignment in consensus structure derivation for ncRNAs including pseudoknots. Being a type of "fold and align" tool, grammar strings' performance relies on the quality of the *ab initio* structure prediction tools. In the worst case, there is no grammar pattern that can be shared by at least two input sequences, rendering low accuracy of structure derivation. Thus, grammar string-based alignment works best when a few input sequences have consistent structure predictions.

One future direction is to reduce the running time of grammar string comparison. In particular, we need to have a more efficient algorithm to choose input to multiple sequence alignment from hundreds of sub-optimal structures. Second, we need to improve the quality of grammar string

alignment when the *ab initio* structure prediction tools have poor accuracy. One alternative method is to apply simplified SanKoff algorithms to conduct pairwise structure derivation. Then we will use these structures instead of output of existing MFE-based tools as input to grammar string construction.

Chapter 4

Shape and secondary structure prediction for ncRNAs including pseudoknots based on linear SVM

4.1 Background

There are 26,704 sequences in 71 ncRNA seed families of Rfam 10.0[68] containing pseudoknots. With advances in sequencing technologies and structure prediction, more pseudoknot structures are expected to be disclosed.

Many computational methods have been used to determine the native structure of ncRNAs. A native structure is a structure that forms conformationally folding in native state before forming the tertiary structure. The gap between the free energy of the native state and other non-native structures is often small[20]. Thus, misfolded conformations can form with high probabilities[21]. For a review of available tools, please see[69, 70].

Although there is promising progress, finding the native secondary structure is still difficult. In particular, identifying the pseudoknot, an important structural motif in many types of ncRNAs, poses a great challenge for existing methods. Predicting the minimum free energy secondary structure that includes pseudoknots has been proven to be NP-hard [23]. One recent attempt is

to first predict the abstract shapes (or shapes for short), which retain adjacency and nesting of structural features but disregard the length details of helix and loop regions [71]. The predicted shape will then be used to guide structure prediction. The idea of abstract shapes has long been used to characterize different types of structures. For example, most tRNAs have the clover-leaf structure; most pre-miRNAs have the stem-loop structure; many types of pseudoknots have an H-type structure.

While the size of the folding space of an RNA sequence increases exponentially with the sequence length [1], many possible folding only differ in the details of the loop and helix regions and hence have the same abstract shape. Previous analysis shows that the space of the abstract shapes is significantly smaller than the complete folding space [72]. Knowing the abstract shape can significantly reduce the search space for structure prediction tools and improves the accuracy of structure prediction [71, 73]. The utilities of abstract shapes have been demonstrated in a number of recent publications. The Giegerich group used abstract shapes in comparative structure prediction in pseudoknot-free sequences [73]. People use shapes to aid miRNA precursor prediction in large-scale studies [74, 75]. Furthermore, shapes are used to index fast-expanding ncRNA families in Rfam [68] and lead to efficient known ncRNA search [76].

Previous work focused on shape derivation and usage for pseudoknot-free ncRNAs. There is a lack of studies of the usage of shapes in pseudoknot structure prediction. In this work, we predict the consensus shape of a group of homologous ncRNAs that may contain pseudoknots. In addition, we develop a program that uses the consensus shape for consensus pseudoknot structure prediction. A majority of existing pseudoknot structure prediction tools often have topology restrictions such as H-type, recursive H-type [77, 78, 79, 80], kissing hairpin, or complexity levels of pseudoknot using genus numbers [81]. Therefore, using the predicted abstract shapes of input sequences can help remove the topology restriction and leads to more general and practical pseudoknot structure

prediction tools. Compared with existing tools, our tool has the following properties:

- While most existing shape prediction tools use a single sequence as input, we conduct comparative shape prediction on homologous ncRNAs that might contain pseudoknots. Experiments show that comparative structure or shape prediction, which derives the consensus structure or shape from a group of homologous sequences, can achieve better accuracy than using a single sequence [69, 73, 82].

- We can predict the abstract shapes of both pseudoknot-free and pseudoknot-containing sequences.

- Current tools use the shape probability [83] or the sum of energies of structures to rank shapes. We use multiple features by combining well-studied feature ranking methods and the support vector machine (SVM) method.

- We demonstrate the usage of the shape by applying it to pseudoknot structure prediction. The whole software package can be directly used to derive the consensus secondary structure of homologous ncRNAs. The consensus shape prediction tool named KnotShape and the corresponding consensus pseudoknot prediction tool named KnotStructure are publicly available at our website.

We tested our software on hundreds of RNA sequence sets. The experimental results show that we can achieve 18% higher accuracy than the state-of-the-art consensus shape prediction tools on pseudoknot free sequences. For pseudoknot-containing sequences, we achieve approximate 29% higher sensitivity and 10% higher positive predictive value in structure prediction than similar tools.

4.2 Related work

Computational structure prediction can be divided into de novo structure prediction and comparative structure prediction, which derive structures from a single sequence and multiple homologous ncRNAs respectively. As our method is to derive the consensus shape and structure of homologous ncRNAs, we briefly introduce related work in comparative ncRNA structure derivation. There are three general approaches for structure derivation from multiple sequences: simultaneously align and fold, align-then-fold, and fold-then-align. It is computationally expensive to simultaneously align and fold pseudoknot structures. The performance of the align-then-fold pseudoknot prediction heavily depends on the quality of the alignment. Usually multiple sequence alignment (MSA) tools such as ClustalW [51] are used to generate the alignment as the input to the folding tool. However, common structures can be missed due to misalignment between sequences lacking significant similarity [63]. In this work, we design a pseudoknot prediction tool using the fold-then-align strategy that does not require an alignment as input. Tools based on fold-then-align use a de novo folding tool to construct a small but representative sample of the folding space, which consists of the predicted optimal and sub-optimal structures. Structures from the folding space are chosen to maximize the structural and sequence similarity.

A number of software packages exist to predict the abstract shape for a single sequence. The sum of energies or the accumulated Boltzmann probabilities of all structures within a shape have been used as main features for shape prediction. The latter is often referred to as the shape probability. Usually the shapes with small sum of energies or high shape probabilities are more likely to be the correct shapes. It is claimed in RapidShapes [83] that using shape probabilities has superior performance over free energy-based approach because of its independence on sequence length and base composition. However, exact computation of the shape probability incurs exponential com-

putational cost to the sequence length [83]. Thus, various heuristics or restrictions [84, 85] have been adopted for fast shape probability computation.

RNAcast [73] derives the consensus shape from homologous pseudoknot-free sequences based on the fold-then-align strategy. Structures are grouped based on their shapes and shapes are ranked by sum of free energies of structures within the shape in ascending order. The first-ranked shape is presented as the consensus shape. The consensus structure is derived from the lowest free energy structures of each sequence within the shape.

4.3 Methods

4.3.1 RNA structures and their representations

4.3.1.1 RNA structures and pseudoknots

RNA molecules fold into complex three dimensional structures by nitrogenous bases that are connected via hydrogen bonds [86] (Figure 1a). The secondary structure of an ncRNA is defined by the interacting base pairs. Some RNA molecules fold into pseudoknot structures by paring bases in loop regions with bases outside the stem loop (Figure 1b).

In this work, two types of ncRNA secondary structure representations are used. The first type is the arc-based representation, where nucleotides and hydrogen bonds are represented by vertices and arcs, respectively (Figure 1c). For pseudoknot-free secondary structures, all arcs are either nested or in parallel. Crossover arcs indicate pseudoknots. The second type is based on dot-bracket notation, where '.' represents unpaired bases and matching parenthesis '(' and ')' indicate base-pairing nucleotides. Following the annotation of Rfam [68], we use an extended dot-bracket notation to represent pseudoknot structures. The base-pairing nucleotides forming pseudoknots

are represented by upper-lower case character pairs, such as A..a or B..b, as shown in Figure 1d.

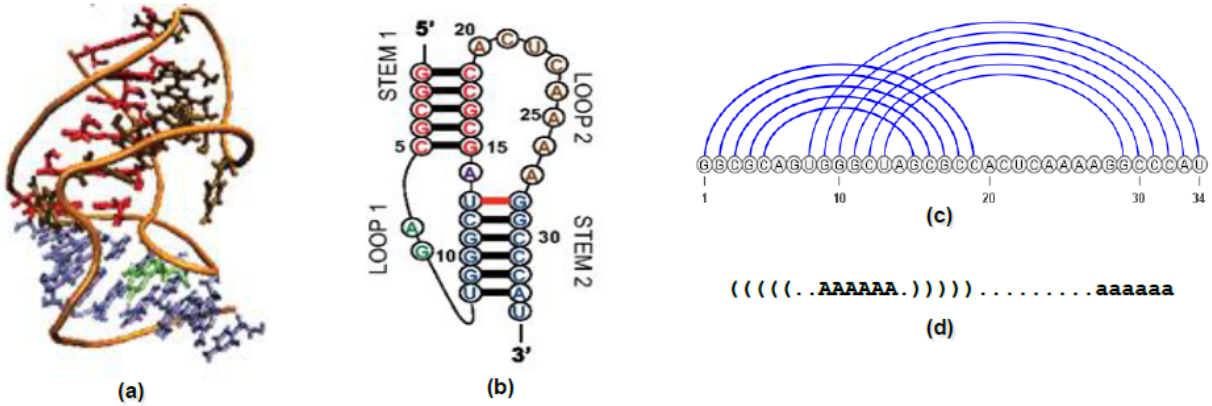


Figure 4.1 Structure of an RNA pseudoknot. (a-d) show the three-dimensional structure, secondary structure, arc-based representation, and dot-bracket notation of mouse mammary tumor virus (MMTV) H-type pseudoknot with PDB code 1RNK. The bases in stacking regions are colored with red and blue while the unpaired bases are colored with green and brown.

4.3.1.2 Abstract shapes

Abstract shapes were formally introduced by Giegerich et al [71]. The folding space of a given RNA sequence is partitioned into different classes of structures, by means of abstracting from structural details. These classes are called abstract shapes, or shapes for short. An RNA secondary structure can be mapped to an abstract shape with different levels of abstraction [73]. In the abstract shape, details about the lengths of the loop and stacking regions are removed (see Figure 1 for examples of stacking and loop regions). Stacking regions are represented by pairs of brackets and unpaired regions are represented by underscores.

Pseudoknots are represented by pairs of upper-lower case characters. Figure 2 presents examples of the abstract shapes of level 1, 3, and 5 of a pseudoknot-free structure and a pseudoknot. Level 5 represents the strongest abstraction and ignores all bulges, internal loops, and single-stranded regions. Level 3 adds the helix interruptions caused by bulges or internal loops. Level 1 only abstracts from loop and stack lengths while retains all single-stranded regions.

(a)
```
AUCGGCGCACAGGACAUCCUAGGUACAAGGCCGCCCGUU
..(((.((..(((....))).(((.....)))))))..
L1: _[_[_[_]_[]]]_
L3: [[[][]]]
L5: [[][]]
```

(b)
```
GGCGCAGUGGGCUAGCGCCACUCAAAAGGCCCAU
(((((..AAAAAA.)))))........aaa.aaa
L1: [_AA_]_a_a
L3: [AA]aa
L5: [A]a
```

Figure 4.2 Examples of abstract shapes in level 1, 3 and 5. (a) The abstract shapes of a pseudoknot-free structure. (b) The abstract shapes of a structure with a pseudoknot.

4.3.2 Shape prediction

In this section we describe KnotShape, a comparative shape prediction tool for homologous ncRNA sequences that allows pseudoknots. The task of shape prediction is to select the best representative shape for given homologous sequences. In order to identify the best shape, various features such as shape probability [83], sum of energies of all structures in this shape [73], and the rank sum score [73] are evaluated to rank shapes. It has not been systematically assessed whether combinations of multiple features can lead to better shape prediction. In this work, we incorporate Support Vector Machine (SVM)[87] and feature selection techniques to determine important features for shape identification. In addition, we adopted a machine learning-based scoring function to evaluate the qualities of shapes.

The method contains two important components. The first one is the consensus shape prediction (KnotShape) and the second one is structure prediction using predicted shape as input (KnotStructure). We will first describe KnotShape, focusing on the feature construction and selection strategy. Then we will describe how to derive the consensus structure given the consensus shape.

4.3.2.1 Notation

Figure 4.3 illustrates the mapping between sequences, structures, and shapes. The input is a set of homologous ncRNAs and the output is the predicted consensus shape. Notations used in this paper correspond to this mapping.

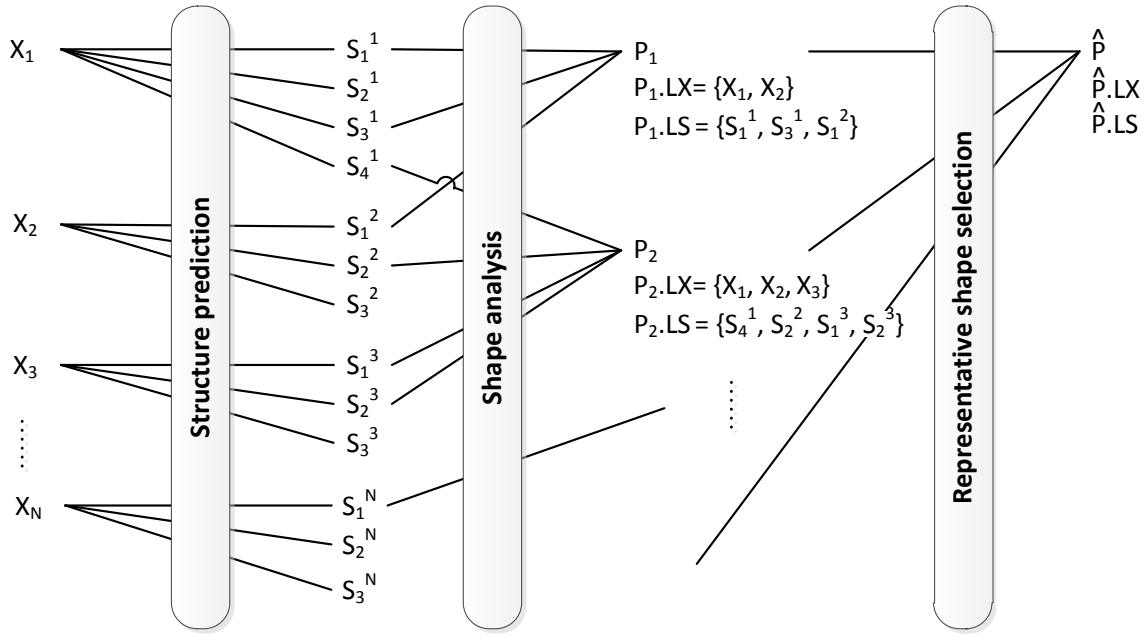


Figure 4.3 The relationship between sequences, structures, and shapes.

- The N homologous ncRNAs constitute the input sequence space. X_i represents the ith sequence.

- Each sequence can be folded into different secondary structures. Let S^i represent the set of folded structures of the ith sequence X_i. The set of structures predicted from all N input sequences is the union of S^i: $S = S^1 \cup S^2 \cup \ldots \cup S^N$.

- S^i_j is the jth structure in the folding space of X_i. Its free energy is denoted by $\Delta G(S^i_j)$. For a sequence X_i, the minimum free energy $\text{MFE}(X_i)$ is the lowest free energy among the energies of all predicted structures of X_i, i.e. $\text{MFE}(X_i) = \min_{1 \le j \le |S^i|} \Delta G(S^i_j)$.

- All structures in S can be classified into a set of abstract shapes. For a shape P, we record its associated sequences and structures. *P.LX* denotes the set of associated sequences, each of which can fold into a structure with shape P. *P.LS* denotes all structures with shape P.

- $\hat{P}$ is the predicted shape of the given homologous sequences $X_1, X_2, .., X_N$.

In order to explore the large folding space of multiple homologous sequences, we use a de novo folding tool to output the optimal and sub-optimal structures within a given energy cutoff. This heuristic does not allow us to explore the complete folding space. Given the observation that the correct structure is usually close to the optimal structure, this heuristic works well in practice [88].

4.3.2.2 Feature construction and selection

Intuitively, the correct shape tends to possess the following properties. The correct shape should have high shape probability, meaning that a large number of structures can be classified into this shape. When we have multiple homologous sequences as input, the correct shape should be well-represented by all or a majority of the input sequences. Also, the ranking of the structure with the correct shape in the folding space of each sequence should be high. In addition, some structures with the correct shape have low thermodynamic energies. For the energy-related properties, various measurements can be introduced. For example, instead of using the sum of the energies of all structures within a shape, one can use the smallest energy. Furthermore, more complicated properties such as the sequence similarity for all sequences associated with a shape P and the structural similarity of structures associated with a shape P might contribute to the shape prediction too. These similarities can be quantified using different methods such as k-mers profiles, multiple sequence alignment scores, variation of base pairs and so on.

It is not trivial to decide whether a single property is enough to choose the correct shape. If not, which combination of these properties can lead to the best shape prediction performance. In order to systematically choose a set of features (i.e. properties) for shape prediction, we use F-score [89] to measure the discrimination between a feature and its label. Given the feature vector x_k,

$k = 1, .., m$, the F-score of the ith feature is defined as:

$$F(i) \equiv \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{i=1}^{n_+}(x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n_- - 1}\sum_{i=1}^{n_-}(x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2}$$

where n_+ and n_- are the numbers of positive and negative instances respectively. $\bar{x}_i$, $\bar{x}_i^{(+)}$, and $\bar{x}_i^{(-)}$ are the average values of the ith feature of the whole, positive labeled, and negative labeled data. $x_{k,i}^{+}$ and $x_{k,i}^{-}$ are the values of ith feature of the kth positive and negative instances respectively.

F-score reflects the discrimination of the features. The higher the F-score, the more discriminative the feature is. F-score is known to have a disadvantage in that it does not carry out the mutual information between features as it considers each feature separately. However, F-score is simple and quite effective in practice.

Feature selection searches for the optimal subset of features [90]. There exist different methods for feature selection. In this work, we adopt sequential forward search (SFS) [91] because of its simplicity and effectiveness. Starting with an empty set, we iteratively select one feature at a time and add it to the current feature set. Features are selected in a descending order of the discriminative power determined by the F-score. The feature added is the one that gives the highest accuracy.

Based on the properties that might be relevant to consensus shape prediction, we construct 17 features and compute the F-score for each of them. The accuracy is evaluated using a linear SVM method. The standard grid search approach is used to find an optimal SVM parameter. The performance of a feature set is evaluated using 5-fold cross validation. Prediction accuracy is the average value of all cross validation sets. The feature set that achieves the highest accuracy includes the following four features.

- *F1: the contribution of sequences*. We capture the contribution of sequences using the number of sequences mapped to the shape. This feature reveals how the shape is shared among

the homologous sequences. $F1 = |P.LX|$.

- *F2: the contribution of structures*. This feature represents the abundance of structures mapped to the shape. $F2 = |P.LS|$

- *F3: the average free energy*. Energy model is commonly used to determine the stability of predicted structures. The basic idea behind this feature is that a stable shape is expected to be derived from a group of stable structures. $F3 = \frac{\sum_{S \in P.LS} \Delta G(S)}{|P.LS|}$.

- *F4: the average of minimal free energy*. This feature is different from $F3$ in that it considers only the minimal free energy among all predicted structures of each sequence. $F4 = \frac{\sum_{X \in P.LX} MFE(X)}{|P.LX|}$.

4.3.2.3 Shape ranking using a simple scoring function

Once the features are determined, they are used together with a trained linear SVM for shape labeling. Multiple shapes might be labeled as "true". In order to rank these candidate shapes for the final shape selection, we evaluate each candidate shape using a score named *sc*, which is proportional to the signed distance between the candidate shape to the classification hyperplane [92]. Specifically, $sc = w \cdot x + b$, where $\cdot$ denotes the dot product, w is the weight vector, and x is the instance vector. w is trained on the optimization function in the linear SVM. The larger $|w_j|$ is, the more important the jth feature is. This is restricted to w in a linear SVM model.

4.3.2.4 Time complexity of shape prediction

For N input sequences, there are S predicted structures. These structure can be grouped into P' shapes. As we use the de novo folding tools to output near-optimal structures within a given energy range (e.g. 5%), we found that $N : S : P' \approx 1 : 10 : 1.375$. Mapping structures to shapes takes $O(SL)$,

where L is the sequence length. As sorting shapes according to their features takes $P'log(P')$ and $P' \leq 2N$ and $S \leq 11N$, the procedure of shape prediction has time complexity $O(NL+NlogN)$.

4.3.3 Consensus structure prediction given a shape

Once we determine the shape, we will predict the structure in the shape class for the given homologous ncRNAs. Structures corresponding to the same shape can differ significantly in the details of the loop and stacking regions. A strategy is needed to choose the correct structure inside the shape class for each input sequence. The simplest strategy is to output the MFE structure for the chosen shape, which has been used in previous work [73]. However, the MFE structure in a shape may not be the native structure. In particular, the accuracy of the MFE prediction for ncRNAs containing pseudoknots is low.

In this section we describe KnotStructure, a comparative structure prediction method for homologous sequences given the shape. The rationale behind comparative structure prediction is that the secondary structures and sequences are conserved during evolution. Thus, finding the structures to maximize both the sequence and the secondary structure similarity among homologous ncRNAs provides the basis for comparative structure prediction. Various methods for evaluating structural and sequence similarity exist. The major challenge is to efficiently select $|\hat{P}.LX|$ representative structures to achieve the highest structural and sequence similarity.

As we already derived the consensus shape $\hat{P}$ using KnotShape, only structures with shape $\hat{P}$ will be allowed. In addition, for each sequence $X_i \in \hat{P}.LX$, only one structure with shape $\hat{P}$ can be selected. The total number of combinations of structures for measuring the similarity is thus $\Pi_{i=1\ to\ |\hat{P}.LX|}|\hat{P}.LS \bigcap S^i|$, where $\hat{P}.LS \bigcap S^i$ contains structures with shape $\hat{P}$ for a sequence X_i. Although efficient heuristics exist to measure the similarity among multiple structures and sequences, the sheer amount of combinations poses a great computational challenge.

Procedure 1 Representative structures selection

Input: $\hat{P}$, $\hat{P}.LX$, $\hat{P}.LS$
Output: The representative structures
1. Initialization
for Every two structures S_i^x and S_j^y **do**
//only evaluate similarity of structures from different sequences
if $x \neq y$ **then**
Evaluate the similarity of S_i^x and S_j^y
else
S_i^x and S_j^y has similarity $-\infty$
end if
end for
2. Select the set of representative structures using hierarchical clustering
//Each structure is a cluster by itself
repeat
Combine a pair of clusters with the highest similarity
For any structure S_i^x added to the cluster, remove all other structures S_j^x for $j \neq i$
Re-evaluate the similarity between clusters
until the cluster has size $|\hat{P}.\text{LX}|$

In order to efficiently select representative structures, we use a similar method to Unweighted Pair Group Method with Arithmetic Mean (UPGMA), an agglomerative hierarchical clustering technique [93]. Each object (i.e. secondary structure) starts in its own cluster. The closest pair of clusters is selected and merged into a single cluster as one moves up the hierarchy. The distance between clusters is measured using arithmetic mean defined in UPGMA. Compared to the standard clustering procedure, we have constraints on the objects that can be selected into the same cluster. Given the shape, only structures that have shape $\hat{P}$ and come from different ncRNAs can be combined in the same cluster. The detailed clustering process is described in Procedure 1.

During clustering, the structural and sequence similarity is evaluated using grammar string-based approach [94, 95]. Grammar strings encode both secondary structure and sequence information for an ncRNA sequence. Grammar string alignment score can accurately quantify the structural and sequence similarity of two ncRNAs. In addition, grammar string can encode pseudoknot structures [94, 95]. For a sequence X_i and one structure S_j^i in the folding space of X_i, X_i

and S^i_j are encoded in a grammar string gs^i_j. We measure the similarity between any two grammar strings using the normalized grammar string-based alignment score over the alignment length. The similarity between groups of grammar strings is measured by arithmetic mean in UPGMA.

Figure 4.4 sketches the representative structure selection based on clustering procedure. Let gs^i_j be a grammar string converted from X_i and S^i_j, $X_i \in P.LX$. Once gs^i_j is selected, all the other grammar strings derived from the folding space of X_i will be removed from further processing.

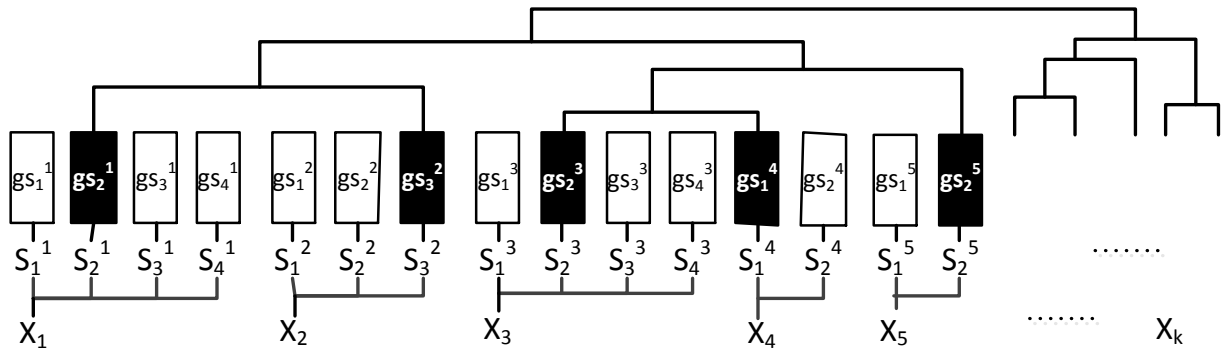


Figure 4.4 An example of structure selection based on hierarchical clustering. For each structure S^i_j in the folding space of sequence X_i, the grammar string encoding the structure and the sequence is denoted as gs^i_j. All sequences and their associated structures are converted into grammar strings before clustering. The highlighted rectangles indicate grammar strings that are selected as representative structures.

The progressive MSA is performed on the set of representative structures using the clustering path as a guide tree. We then derive the consensus secondary structure from the alignment. The consensus structure can be mapped to each aligned sequence to accomplish the predicted structure of an individual sequence.

4.3.3.1 Running time of structure prediction

Converting a sequence and an associated secondary structure into a GS (grammar string) takes $O(L^2)$, where L is the length of the sequence. Let the number of structures in $\hat{P}.LS$ be m. It takes $O(L^2 m)$ to encode all structures with shape $\hat{P}$. In the first step of hierarchical clustering, we

measure the similarity between GSs of different ncRNAs by conducting all-against-all comparison. Conducting pairwise GS alignment takes $O(l^2)$, where l is the length of the GS sequence and $l \leq L$. By using the default energy cutoff (5%) for sub-optimal structure generation, we observed that $m \leq 11N$. Thus, the all-against-all similarity measure has time complexity $O(L^2N^2)$. The guide tree generated using the clustering procedure contains at most N representative structures. Thus, the total running time for clustering is $O(L^2N^3)$, which is the leading time complexity term for the consensus structure prediction algorithm.

4.4 Results

4.4.1 Data sets

The training data set is the *K10* from BaliBASE [40]. It contains 845 sequence sets, each of which has 10 homologous ncRNAs. There are two test data sets. The first one is the *K15* from BaliBASE. *K15* contains 503 sequence sets, each of which has 15 homologous ncRNAs. As existing shape prediction tools are not designed for handling pseudoknots, we use the pseudoknot-free sequence sets in *K15* to compare the performance of shape prediction. After removing the sets containing pseudoknots, we have 452 sequence sets left. To test the performance of pseudoknot prediction, we constructed the second test set *R15* from psuedoknot families of Rfam [68]. In Rfam 10.0, there are 71 families containing pseudoknots. 25 of them have published structures. Of the 25 families, only families with at least 15 seed sequences are used for testing our tools. For each chosen family, sets of 15 sequences are chosen randomly to construct the test sets. Finally *R15* contains 160 test sets. The average pairwise sequence identities range from 60-93%. For all sequence sets, the reference shapes were derived from Rfam [68].

4.4.2 SVM training

For both the training and testing data sets, we need to apply de novo folding tools to the sequences. We choose a folding tool using the following criteria. First, this tool is able to output both the optimal and sub-optimal structures. Second, this tool has high accuracy and can be efficiently applied to a large number of ncRNAs. Finally, if the target sequences contain pseudoknots, this tool should be able to output pseudoknot structures. As a result, we chose TT2NE [81]. Different from many other pseudoknot prediction tools that have constraints on the type of the pseudoknot, TT2NE is more flexible about the types of the target sequences. However, when it was applied to *K10*, TT2NE failed to output structures for some sequences due to the length limit (200 nt) and also existence of IUPAC characters in some sequences. Thus, for the training data set *K10*, we applied quikfold [15] because *K10* rarely contains pseudoknots. Although it is ideal to use the same folding tool to the training and testing data set to achieve optimal classification performance, the complexity of the training and test data sets together with the performance of de novo folding tools lead to the current combination. In the Discussion Section we will briefly discuss how de novo folding tools affect the performance.

We employed the SVM model implemented by LIBSVM tool [96] for classification. For each sequence in *K10*, we applied quikfold with the energy range 5% to obtain both optimal and sub-optimal structures of each sequence. The predicted structures were grouped based on their corresponding shapes. Associated features were extracted and enclosed with each shape. We normalized feature values to fit the different properties of test sets to the same scale.

To label shapes, we used the shapes extracted from the consensus structures in Rfam [68] as the reference. Shapes are labeled according to their correctness. We label a shape as 1 if it is as same as the reference shape. Otherwise, it is labeled as -1.

4.4.3 Shape prediction comparison

We compared KnotShape with RNAcast [73], which is part of RNAshapes package [84]. RNAcast takes the sequences as the input and predicts the consensus shape shared by all sequences. As it is not designed for pseudoknot structures, we only applied RNAcast to 452 test sets of *K15*, which are pseudoknot-free. TT2NE is applied to the test set using the default parameters. For each sequence, the optimal structure and 10 sub-optimal structures are kept as the sample of the folding space for each sequence. We compared our predicted shapes and the first-ranked shapes output by RNAcast with the reference shapes derived from Rfam [68]. The comparison is presented in Table 4.1. Note that RNAcast cannot output the shapes containing pseudoknots and thus is left blank for *R15* in Table 4.1. The accuracy of KnotShape is 18% higher than RNAshapes.

Table 4.1 Accuracy of shape predictions

	K15			*R15*		
	Testset	Correct shapes	%Accuracy	Testset	Correct shapes	%Accuracy
KnotShape	452	311	68.81	160	107	66.88
RNAcast	452	232	51.33	-	-	-

4.4.4 Structure prediction comparison

We applied the predicted shapes to pseudoknot structure prediction and compared the structure prediction performance with IPknot [80], HxMatch [97], and TurboKnot [82], which are chosen because of their popularity, availability, and easy usage on large number of sequences. Sequence alignments were generated using ClustalW and entered as the input to IPknot and HxMatch. For IPknot, we chose the appropriate levels of prediction according to the test sets. We ran Hxmatch with the default parameters. We used the parameters suggested in [82] to run TurboKnot. Sensi-

tivity and the Positive Predicted Value (PPV) are used to evaluate the performance:

$$Sensitivity = \frac{TP}{TP+FN}, PPV = \frac{TP}{TP+FP}$$

TP is the number of correctly predicted base pairs. *FN* is the number of base pairs that are in the reference structure but not in the predicted structure. *FP* is the number of base pairs that are in the predicted structure but not in the reference structure. Figure 4.5 is the boxplot of the sensitivity and

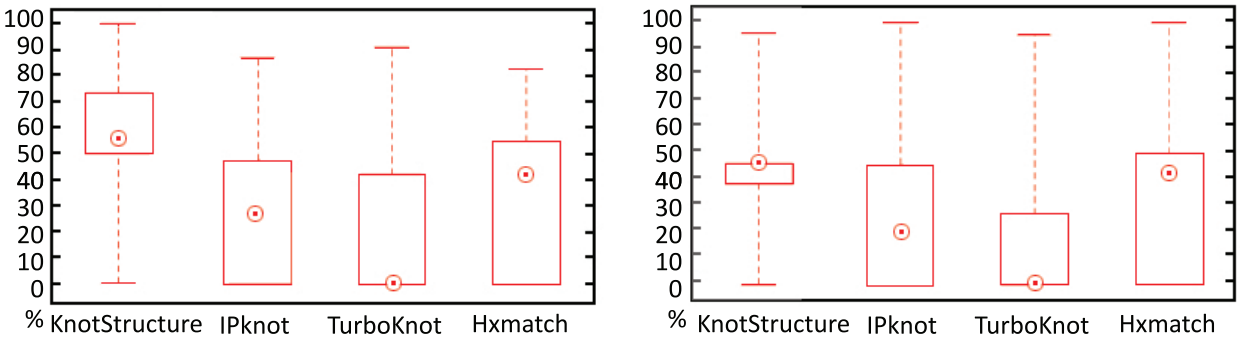


Figure 4.5 Comparison of the sensitivity and PPV of different tools.

PPV over all 160 test sets. KnotStructure has the best overall performance on the whole data set. The median values of sensitivity and PPV are 54.55% and 46.15% for KnotStructure. Hxmatch has the next highest sensitivity and PPV (42.11% and 42.86% respectively). The abstract shapes of these families are shown in Table 4.2. Three families contain simple H-type pseudoknots while the other three families contain more complicated pseudoknots. In order to show the effect of shape prediction in structure prediction, we predicted the structures of *R15* using 10 randomly selected shapes. The average sensitivity and PPV of predicted structures with the predicted shapes are higher than those using random shapes as shown in Table 4.3. Table 4.4 and 4.5 show the average sensitivity and PPV over all sequences of each family compared to other tools. The average running time of KnotStructure on each family compared to other tools is shown in Table 4.6.

Table 4.2 Abstract shapes of ncRNA families in *R15*

RNA Type	Shape Level 5	Shape Level 3
HDV_ribozyme	[A[B]]b[]a	[AA[B]]b[[[[[[]]]]]]aa
Alpha_RBS	[ABC]bac	[[[ABC]]]bac
Tombus_3_IV	[[]A][][]a	[[[]A]][][]a
Tymo_tRNA-like	[][][]A[a]	[][][[]]AA[aa]
Corona_FSE	[A]a	[AA]aa
Prion_pknot	[A]a	[A]a

Table 4.3 Sensitivity and PPV of predicted structures using the predicted shapes and randomly selected shapes

	Predicted shape	Randomly selected shape									
		1	2	3	4	5	6	7	8	9	10
SEN	79.00	45.68	58.41	53.09	58.06	55.74	45.88	44.75	56.12	61.08	46.54
PPV	67.10	38.81	50.92	42.82	49.15	46.24	40.29	36.33	45.75	52.80	38.53

Table 4.4 Sensitivity for different ncRNA families

			Sensitivity			
RNA Type	Len$^{\oplus}$	TS^{*}	KnotStructure	IPknot	TurboKnot	Hxmatch
HDV_ribozyme	89.70	12	**82.52**	<u>50.66</u>	36.47	23.51
Alpha_RBS	110.99	18	**74.36**	<u>46.59</u>	46.24	24.49
Tombus_3_IV	91.61	4	**84.00**	65.91	72.00	<u>80.00</u>
Tymo_tRNA-like	85.12	3	**96.79**	76.41	<u>83.52</u>	75.09
Corona_FSE	82.91	9	**96.14**	56.55	56.55	<u>73.27</u>
Prion_pknot	40.46	114	**40.17**	13.70	2.96	<u>30.26</u>

Bold number and underlined number indicate the highest and the second highest sensitivity for each family.
$^{\oplus}$ Average sequence length. * The number of test sets.

Table 4.5 PPV for different ncRNA families

			PPV			
RNA Type	Len$^{\oplus}$	TS^{*}	KnotStructure	IPknot	TurboKnot	Hxmatch
HDV_ribozyme	89.70	12	**80.13**	<u>59.24</u>	39.28	39.77
Alpha_RBS	110.99	18	**40.59**	<u>25.34</u>	23.70	22.19
Tombus_3_IV	91.61	4	<u>83.14</u>	78.02	73.47	**90.91**
Tymo_tRNA-like	85.12	3	**90.25**	73.62	86.85	<u>90.11</u>
Corona_FSE	82.91	9	<u>75.65</u>	60.51	42.19	**92.30**
Prion_pknot	40.46	114	**32.85**	15.71	3.03	<u>28.86</u>

Bold number and underlined number indicate the highest and the second highest PPV for each family. $^{\oplus}$ Average sequence length. * The number of test sets.

Table 4.6 Running time for different ncRNA families (seconds)

RNA Type	KnotStructure	IPknot	TurboKnot	Hxmatch
HDV_ribozyme	1.55	0.35	28.22	0.41
Alpha_RBS	1.93	0.49	37.25	0.55
Tombus_3_IV	1.80	0.37	13.66	0.42
Tymo_tRNA-like	1.78	0.24	12.24	0.28
Corona_FSE	1.51	0.30	12.75	0.35
Prion_pknot	1.07	0.10	4.23	0.12

4.5 Discussion and conclusion

Based on the fold-then-align strategy, choice of folding tools can play an important role in the performance of the shape and structure prediction. For the test set, we tested two folding tools: HotKnots [98] and TT2NE. We used them in three different ways: Hotknots, TT2NE, and both of them. We ran HotKnots and TT2NE with default parameters. The experimental results show that using TT2NE alone achieves the best performance in consensus structure prediction. It is likely that other folding tools exist to yield better performance than TT2NE. However, as the performance of those tools also depends on the input data and the parameters, a systematic study is needed to choose the best tool.

For TT2NE, we currently only use 10 sub-optimal structures. Increasing this number moderately does not affect the performance significantly. It indicates that the correct structures have high rankings in the folding space. There are more pseudoknot-free structures available than pseudoknot-containing structures. To achieve a reliable SVM model, more training data is desired. We used *K10* for feature selection. This may cause KnotShape to have slightly lower predictive performance on pseudoknot-containing than pseudoknot-free sequences. Nonetheless, the features used in KnotShape does not heavily rely on the free energy value, which is different between pseudoknot-free and pseudoknot-containing structures. Instead, the feature set is based on multiple RNA properties shared among homologous sequences.

Extensive analysis of RNA properties based on SVM allows us to identify important features related to abstract shapes. The combination of mass data analysis and SVM-based feature ranking makes KnotShape a promising tool for shape prediction. By combining the predicted shapes and the multiple structural alignment strategy, KnotStructure demonstrates higher accuracy in pseudo-knot structure prediction.

Chapter 5

LncRNA-ID: Long non-coding RNA IDentification using balanced random forest classification

5.1 Background

Recent study indicates that there are at least four time lncRNA transcripts more than protein-coding transcripts [3]. The majority of lncRNAs are transcribed in the sense and antisense directions and some of those overlap with protein-coding genes. Unlike other ncRNAs such as miRNAs or snoRNAs that are strong conserved across diverse species [99], lncRNAs are poorly conserved [100]. The poor conservation of ncRNAs may be the result of recent and rapid adaptive selection, as evidenced by the existence of many lineage specific ncRNAs, such as Xist or Air [101].

LncRNAs exist in many species such as Arabidopsis [102], Zea mays [103], honey bee [104], chicken [105], zebrafish [106], etc. In recent years, a large number of lncRNAs have been identified. GENCODE [10] comprises 9,277 manually annotated lncRNA genes in the human genome. The LncRNADisease database [107] contains 1,564 human lncRNAs that are likely to be associated with diseases. Thus, given the functional importance and ubiquity of lncRNAs, it is important to annotate them on a genome scale in various species. With the advances of the next-generation

sequencing technologies, the transcriptomes of a large number of organisms have been sequenced, providing us a unique opportunity to mine lncRNAs. The assembled transcripts contain different types of functional elements such as small ncRNAs, lncRNAs, and protein coding genes. As lncRNAs are usually much longer than small ncRNAs, lncRNAs can be effectively distinguished from small ncRNAs using size as the main criteria. However, lncRNAs have similar splicing structures as protein coding transcripts and tend to encode putative open reading frames (ORFs). Thus, a major challenge for lncRNA identification is to distinguish lncRNAs from protein coding genes, especially in non-model organisms lacking comprehensive protein coding gene annotation.

5.2 Related work

Many efforts have been made to distinguish between lncRNA and protein-coding transcripts, ranging from applying a threshold for a single feature to more complicated supervised machine learning methods. One commonly used feature is the length of the ORF. For example, a simple approach is to classify a transcript containing an ORF of length above 100 amino acids as protein-coding gene [108]. This criteria is arbitrary and is not always correct [109]. By using this simple criteria, the mouse Xist RNA gene [110], which encodes a putative ORF of 298 amino acids (aa), was mis-classified as a protein-coding gene when it was first discovered [111].

More accurate approaches for identifying lncRNAs use supervised machine learning methods by formulating lncRNA discovery as a binary classification problem. These approaches can be further divided into two types. One relies on sequence alignments and the other is alignment-free. Representative examples of alignment-based methods include Coding-Potential Calculator (CPC) [112] and Phylogenetic Codon Substitution Frequencies (PhyloCSF) [113]. CPC aligns transcripts against known protein databases while PhyloCSF uses multiple sequence alignments.

As homologous protein-coding genes tend to share higher sequence conservation than lncRNAs, the alignment score or its statistical significance provides useful information to differentiate these two types of transcripts [114, 115, 116]. However, alignment-based methods usually require high quality alignments [117], which are not trivial to produce and can incur high computational cost.

An alternative and faster approach is alignment-free methods such as CPAT [118], which integrates linguistic features of transcript sequences into a logistic regression model for lncRNA prediction. In addition, rather than using the pre-built models, CPAT allows users to create a model with their own data. This option, which is not present in CPC and PhyloCSF, is very useful for lncRNA identification in different species.

Despite the promising progress for lncRNA identification, there is still a need for better approaches and tools. In particular, existing machine learning-based tools do not carefully handle the imbalanced training data, in which one class has far more instances than the other. The issue of imbalanced training data is particularly pronounced for lncRNA identification when it is formulated as a binary classification problem in existing tools. For example, due to poor annotation of lncRNAs, many species have far less characterized lncRNAs than protein-coding genes. As a result, a classifier tends to over-predict query transcripts as the major class [119]. In addition, many existing tools need users to provide a score threshold for lncRNA identification, which is not always obvious from users' perspective. For example, PhyloCSF and CPAT do not suggest the specific type of an input transcript, but only output a coding potential score. The predefined score cutoffs of PhyloCSF and CPC vary from species to species. PhyloCSF's score cutoffs of 50 and 300 were used for mouse [120] and Zebrafish [121], respectively. CPAT suggests the score cutoffs of 0.364 and 0.44 for human and mouse, respectively. These specific score cutoffs cannot be immediately applied to other species. Even worse, not every tool can be trained on different species to provide users necessary information for choosing an appropriate score cutoff.

In this paper, we present LncRNA-ID, a lncRNA identification tool, which applies random forest (RF) classification [122] to distinguish lncRNAs from protein-coding genes. RF is a classification model aggregating multiple classification trees generated from boot-strap samples and has been successfully applied in bioinformatics [123, 124, 125]. LncRNA-ID has several advantages over existing tools. First, it still takes advantage of alignment-based features, which have strong discriminative power. However, instead of using genome-scale multiple sequence alignments or pairwise alignments against all existing protein sequences, LncRNA-ID employs profile hidden Markov model (profile HMM) based alignments, rendering more sensitive homology search and shorter running time than existing alignment-based lncRNA identification tools. Second, LncRNA-ID is easy to use as it does not require users to provide a score cutoff. It automatically determines the type of a query transcript as well as providing a coding potential score. Third, LncRNA-ID can be applied to various species by providing an option to train the classifier for different data. Fourth, LncRNA-ID does not require a large number of training data of neither protein-coding transcripts or lncRNAs to construct a classifier and can handle imbalanced classes in the training data. In our experiments, we evaluated the performance of LncRNA-ID on two different species, human and mouse. It achieved the highest sensitivity and specificity compared with CPC, CPAT and PhyloCSF on both species.

5.3 Methods

In this section, we first talk about the features used in LncRNA-ID and then describe the method we use to construct our classification model. The features used in LncRNA-ID are derived from three different groups: open reading frame (ORF), ribosome interaction, and protein conservation. Each feature is selected either based on the literatures or the empirical observations. Using multiple

features can significantly improve the performance of classification.

5.3.1 ORF features

ORF is one of the most commonly used criteria to distinguish a lncRNA from a coding transcript. A true protein-coding transcript tends to have longer ORFs than those in lncRNAs. We derive two ORF-related features: ORF length and ORF coverage. The ORF length is defined as the length of the longest reading frame identified in three forward frames. The ORF coverage is defined as the ratio of the length of the chosen ORF to the length of the transcript. From our observation, lncRNAs tend to have shorter ORF and lower ORF coverage than protein-coding transcripts.

5.3.2 Ribosome interaction features

These features are based on the interaction mechanism between the ribosome and mRNAs during protein translation [126]. Ribosomes consist of two parts, a large subunit where two tRNA binding sites are located and a small subunit where the mRNA binding site is located. The translation is initiated when the small ribosomal subunit attaches to the mRNA at a start codon. The ribosome starts to translate the mRNA towards 3' end until it encounters a stop codon. At the end of the protein translation, termination factors release the synthesized protein for use in the cell and the ribosome splits back into large and small subunits [127]. Many studies have successfully applied ribosome footprint to identify functional proteins [128, 129, 130, 131]. In particular, ribosome profiling [129], which sequences mRNA fragments bound to ribosomes, provide a quantitative snapshot of protein translation. However, the availability of ribosome profiling data is still limited. Thus, in this work, we design computational features to quantify the main attributes related to ribosome interaction with mRNAs. We define the features according to these interaction states:

initiation, translation, and termination.

5.3.2.1 Initiation:

The initiation interaction features are derived from the Kozak motif. The Kozak consensus is a favorable motif for a ribosome scanning pattern and initiates translation. It greatly impacts protein translation efficiency [132, 133, 134]. Kozak motif has the consensus GCCRCCAUGG (R represents purine) and is located in the region around the initiator codon of an ORF. In the Kozak determining experiment, single base mutants are preformed on mRNAs and the protein productions of the mutant sequences are measured. It has been demonstrated that nearly all ribosomes will initiate at the start codon [135], AUG. The highly conserved nucleotides at positions -3 and +4 (the A of AUG is +1) and -2 and -1 play a major role in the initiation of the translation process.

We thus derive two features from Kazak motif: the nucleotides at the position {-3, +4} and {-2, -1}. The Kozak features determine the potency of a starting site. A strong starting site, which enhances the translation efficiency, occurs when nucleotides at these positions are conserved, whereas a less conservation indicates a weak starting site [135, 136].

5.3.2.2 Translation:

The interaction between the 3' end of rRNAs and mRNA transcripts exhibits changes of binding energy along the transcript. The binding energy consists of the free energy needed to open the binding site and the energy gained from hybridization. We use RNAup [137] to compute thermodynamics of the interaction between the 3' end of 18S rRNA and a transcript.

In order to capture the change of the binding energy, we calculated a series of binding energies between the 3' end of 18S rRNA and a transcript by moving the 3' end of 18S rRNA toward 3' direction of a transcript one nucleotide at a time. Let δ_i be the free energy at position i. Let N_i

denote the number of Watson-Crick base pairs of an interaction starting at position i. The ribosome coverage is thus defined as:

$$\text{Ribosome coverage} = \sum_{i=1}^{L} \{N_i | \delta_i < 0\},$$

where L is the sequence length. The ribosome coverages were computed on three regions: the whole transcript, ORF, and 3'UTR. These three features illustrate the level of ribosome occupancy on a sequence. For protein coding transcripts, we expect to see higher ribosome coverage on the whole transcript and the ORF region.

5.3.2.3 Termination:

We define the ribosome release score (RRS) to capture a termination signal of ribosomes. The RRS takes advantage of the fact that ribosomes are released when reaching a stop codon. As a result, a sharp drop in ribosome occupancy is seen at the start of the 3' UTR of coding transcripts. In contrast, translational termination should not occur in non-coding transcripts [29, 138].

The RRS is laboratorially measured using the quantitative sequences from a deep sequencing of ribosome-protected mRNA fragments called ribosome profiling[129]. Although ribosome profiling is high quality, but it requires experimental data. Therefore, it is currently not widely available. However, it is expected to become more widely available with the demand from research communities and the progress in cost-effective sequencing technologies.

In the absence of a ribosome profiling data, we estimate RRS as the ratio of ribosome coverage in the putative ORF to ribosome coverage in the corresponding 3' UTR.

$$\text{RRS} = \frac{\text{Ribosome coverage of ORF}/\text{length(ORF)}}{\text{Ribosome coverage of 3}'\text{UTR}/\text{length(3}'\text{UTR)}}$$

RRS indicates the relative degree of ribosome occupancy bias at the terminal binding site in a

sequence. True protein coding transcripts are expected to have larger RRS than non-coding transcripts.

5.3.3 Protein conservation features

True protein coding transcripts tend to show better conservation against characterized proteins. We measure the conservation using profile hidden Markov model (profile HMM)-based alignment scores. In particular, we chose HMMER [139] to align a transcript against all available protein families, such as the ones in Pfam [140]. Applying profile-based homology search has several advantages, compared with pairwise alignment methodologies [141]. First, the number of gene families is significantly smaller than the number of sequences, rendering a much smaller search space. For example, there are only about 14,000 manually curated protein families in Pfam [140]. But they cover nearly 80% of the UniProt Knowledgebase [142] and the coverage is increasing every year as enough information becomes available to form new families [140]. As the profile-based homology search tool HMMER is as fast as BLAST [143], using profile-based search provides a shorter search time. In addition, alignments of query sequences against each protein family are independent from each other and thus can be naturally parallelized on high performance computing platforms. Second, previous work [31] has shown that using family information can improve the sensitivity of remote protein homology search [144]. For the transcriptomes of non-model organism, sensitive remote homology search is especially important for identifying possibly new homologs.

Specifically, each transcript is aligned to all protein families using HMMER. We use 0.1 as the E-value cutoff for HMMER. When more than one alignment is generated for a query sequence, the alignment with the best E-value is used. For the chosen alignment for a transcript, we derive the following three features: (i) the score, (ii) the length of the alignment on the profile, and (iii) the

length of the alignment on the query sequence. A true protein-coding transcript is likely to produce an alignment with higher score and longer alignments than lncRNAs.

In total, we extract 11 features: ORF length, ORF coverage, two Kozak-motif related features, ribosome coverage on three regions: transcript, ORF, and 3'UTR, ribosome release score, alignment score, alignment length on the profile HMM and alignment length on the transcript. It is apparent that although each feature exhibits different value distribution for the two types of transcripts, none of the single feature is able to fully distinguish lncRNAs from coding transcripts. Thus, it is important to combine multiple features to maximize discriminative power. We formalize this problem as a binary classification problem where lncRNAs are defined as the positive class and protein-coding transcripts are defined as the negative class. All these features will be incorporated into the chosen classification model: balanced random forest, which we will describe below.

5.3.4 Balanced random forest

A decision tree is a commonly used classification model in machine learning. Random forest (RF) consists of multiple decision trees. Each decision tree is built from a bootstrap sample, which is a random sample drawn from the training data. During prediction, RF outputs the class agreed by most of the individual trees. We select the RF for the following reasons:

1. It is able to effectively handle missing data, which is common in lncRNA identification. For example, lncRNA transcripts are not likely to have protein conservation and the features such as alignment score or alignment length could be missing.

2. It natively supports categorical features without requiring any transformation. The typical conversion for categorical data is to create dummy binary variables to represent each category value. However, this may decrease the predictive power of the features and is time-

consuming because of the potentially large number of dummy features. With RF, we are able to directly use Kozak motif features without the need for any conversion.

Inspired by Chen et al. [145], we extended RF to balanced random forest (BRF), which contains multiple RFs where each RF is built from a subset of the training data. BRF provides LncRNA-ID with the major advantage that it can learn from the imbalanced training data where the numbers of lncRNA and protein coding samples are highly different. Imbalanced training data is common for lncRNA identification. A recent study found that lncRNAs are at least four times more than protein coding genes in the human genome [3]. In practice, the majority class in the training data is protein-coding transcript because there are more protein coding gene annotation than lncRNA annotation for most organisms. For example, in the GENCODE database [10], there are 12,526 annotated lncRNAs and 95,099 annotated coding transcripts in the human genome. For the mouse genome, there are 6,053 annotated lncRNAs and 47,394 annotated coding transcripts in GENCODE. Thus, there is a need for a classification method that can effectively learn from imbalanced training data where one of the two classes has more samples (majority) than the other class (minority).

When learning from imbalanced training data, there is a high possibility that a bootstrap sample contains very few or even none of the entities in the minority class, resulting in a classification tree with poor performance for predicting the minority class. A naive solution is to either conduct prior over-sampling of the minority class or prior down-sampling of the majority class. Down-sampling usually has a better performance over over-sampling [146]. However, a prior down-sampling of the majority class may result in loss of information, as a large part of the majority class is not used. In contrast, LncRNA-ID employed BRF which ensembles multiple RFs. Each RF was trained by a subset of the majority class and a full set of the minority class. This allows us to achieve better classification performance by maximizing the benefits of using abundant protein-coding and deficient lncRNA data.

Our BRF is different from [145] in that instead of creating balanced training subsets using random drawings, we divide the majority class into equal subsets according to the imbalanced ratio, which is the ration of the size of the majority class to the size of the minority class. The purpose is to maximize the predictive power by ensuring that all training data are incorporated in constructing the classification model. The balanced random forest learning algorithm is shown below:

Procedure 2 Balanced random forest learning

Input: lncRNAs (P) and protein-coding transcripts (N)
Output: a BRF classifier

$k = \frac{|N|}{|P|}$
Create k non-overlapping subsets, $n_1, n_2, .., n_k$, from N
for i=1 to k **do**
  Train a classifier RF_i to discriminate P against n_i
end for
Return an ensemble of $\forall_{i=1}^{k} RF_i$

To create a balanced training data, down-sampling is performed on protein-coding transcripts, creating approximately an equal number of protein-coding and lncRNA transcripts in each subset p_i. Each training subset is then used to create an individual RF. Finally, we integrate all constructed RF classifiers into the BRF. The BRF classifier is then used to predict the type of a query transcript by aggregating the prediction results of ensemble classifiers.

Integrated with balanced random forest methodology using different types of features, LncRNA-ID has the following advantages: (i) can effectively handle limited or imbalanced learning data, which are commonly found in most species; (ii) incorporates different types of features, minimizing bias from a particular group of features. We employ the random forest classification implemented in Weka [147] software package to construct our classification model. The optimal number of trees used in the random forest classification is determined based on the best performance obtained by 10-fold cross validation.

5.4 Results

LncRNA-ID can be applied to different species, and can achieve robust classification performance with imbalanced training data, which is a commonly seen problem for lncRNA classification. To evaluate the performance of LncRNA-ID, we used two data sets from the human genome and one data set from the mouse genome. The first human data set (H1) and the mouse data set (M) were generated from GENCODE consortium [10] within the framework of the ENCODE project. GENCODE is known to have the most comprehensive annotation of long noncoding RNAs available to date. The second human data set (H2) is CPAT's original data set generated from multiple resources: RefSeq [148], a human lncRNA catalog [149], and GENCODE. We further conducted four additional experiments simulating different imbalanced ratios in the training data to demonstrate that LncRNA-ID was able to maintain robust performance with imbalanced training data.

To quantify the classification performance, we used five standard metrics: sensitivity, specificity, accuracy, false positive rate (FPR), and F-score, which are defined as follows:

$$\text{Sensitivity} = \frac{TP}{TP+FN} \ , \ \text{Specificity} = \frac{TN}{TN+FP}, \text{Accuracy} = \frac{TP+TN}{P+N} \ , \ \text{FPR} = \frac{FP}{FP+TN}$$

$$\text{F}-\text{score} = \frac{2 \cdot \text{Sensitivity} \cdot \text{Specificity}}{\text{Sensitivity} + \text{Specificity}}$$

LncRNAs are regraded as the positive class and protein coding transcripts are regarded as the negative class. TP is the number of correctly classified lncRNAs and TN is the number of correctly classified protein coding transcripts. Sensitivity is the proportion of correctly classified lncRNAs in the set of all lncRNAs. Specificity is the proportion of correctly classified protein-coding transcripts in the set of all protein coding transcripts. Accuracy (a.k.a. positive predictive value) is the ratio of correctly classified transcripts in all predictions. False positive rate (FPR) refers to the portion of falsely classified lnRNAs among all protein coding transcripts. F-score

is the harmonic mean of sensitivity and PPV and hence can be used as a single measure for the overall classification performance.

5.4.1 Performance of different groups of features

Before we evaluated the classification performance of LncRNA-ID, we first evaluated the discriminative power of different types of features. We constructed this experiment using the human data set (H1). The detail about this data set can be found in the next section. We created the classification models with the training data using each individual group of features and the combination of different groups. The performance of each classification model was then evaluated with the test data. The overall performance was measured by the area under ROC curve (AUC). AUC is a commonly used method to evaluates performances at all cutoff points, giving better insight into how well the classifier is able to separate the two classes. The greater the AUC is, the better overall classification performance the classifier achieves. The optimal performance is the best FPR and sensitivity that maximizes the F-score.

Figure 5.1 shows the performance of LncRNA-ID using a single group of features versus multiple groups of features. The three groups of features exhibit highly different performance. The ribosome interaction features have the best discriminative power because they are designed based on the protein translation mechanism. According to Figure 5.1, combination of multiple groups of features leads to better performance than using a single group of features. The best performance comes from the combination of three groups of features, which are thus used in all our experiments.

We compared the performance of LncRNA-ID with three state-of-the-art coding-potential prediction tools: CPC, CPAT, and PhyloCSF. These tools output the coding-potential of a transcript and can be used to classify a query transcript into coding or non-coding sequences. Below we present the experimental results of applying LncRNA-ID and the benchmark tools on three data

sets. For each data set, we introduce the training data, test data, and the important parameters used for each tool. We re-train the classification model in CPAT for different data set to optimize its performance. As CPC and PhyloCSF does not provide the re-training option, we use pre-built models. It is worth noting that there is no intersection between the training data and test data.

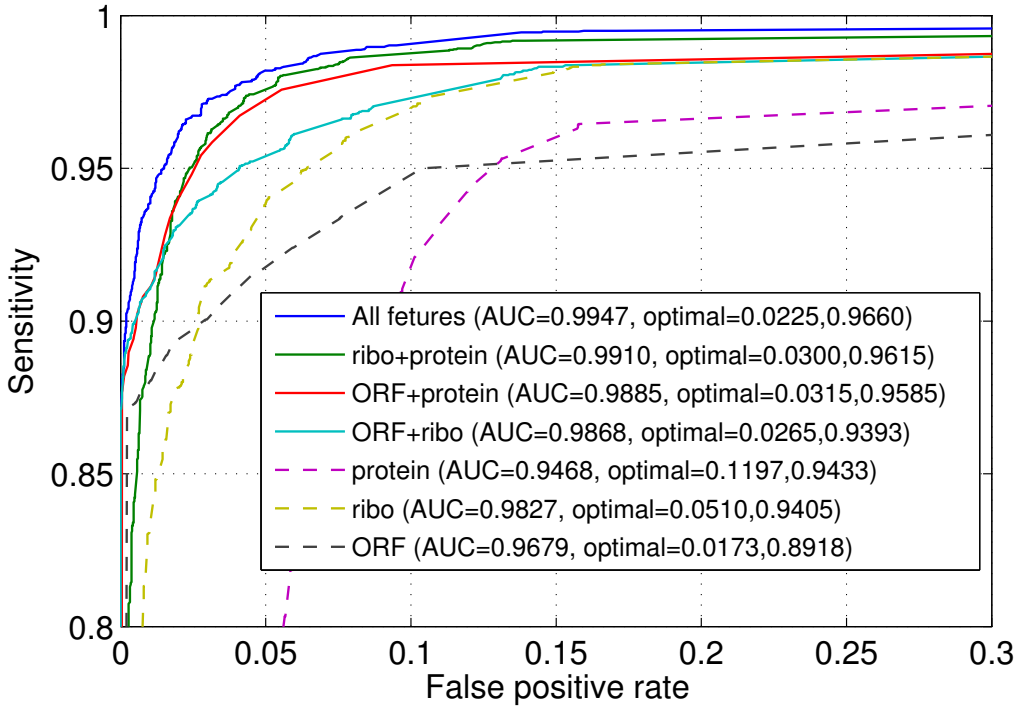


Figure 5.1 Performance comparison among feature groups: ORF features (ORF), ribosome interaction features (ribo), protein conservation features (protein), and the combined feature sets.

5.4.2 The human data set (H1)

This data set contains randomly selected transcripts from GENCODE [10]. The training data contains 48,600 protein-coding transcripts and 8,300 lncRNAs. The test data contains 4,000 coding transcript and 4,000 lncRNAs.

We ran CPC using UniRef90 [150] as the reference protein database, which is a relatively comprehensive protein database suggested by CPC. We created the classification model of CPAT

from the training data using the script provided in CPAT's software package. The created classifier was then used to predict the transcripts in the test data and the performance was evaluated using the CPAT's suggested optimal score cutoff.

A multiple sequence alignment of 45 vertebrate genomes, including the human genome, was downloaded from the UCSC Genome Browser and was used as the input alignment for PhyloCSF. We specified the option that allowed PhyloCSF to search all three reading frames and report the best result as suggested in the PhyloCSF website [151]. We used the default score cutoff of PhyloCSF to generate the classification results.

Table 5.1 shows the comparison of classification performance of all tools on H1. LncRNA-ID had the best sensitivity and accuracy among all tools. Although CPC had the highest specificity, its sensitivity and accuracy were much lower than those of LncRNA-ID. CPC's classifier is based on six features. Three of them are ORF-related features and the others are derived from the alignments of a query sequence against existing protein sequences. These features could cause a bias toward protein-coding transcripts if a lncRNA contains an ORF sharing similarity with existing protein sequences. This might be a major reason behind the low sensitivity of CPC. LncRNA-ID also had the highest F-score and classification accuracy among all tools. Therefore, LncRNA-ID demonstrated the best overall performance in distinguishing protein coding transcripts from lncRNAs among all tools.

The values of AUC for all ROC curves can be found in Figure 5.2. LncRNA-ID also had the best AUC among all the tools. In addition, we gave the sensitivity and FPR corresponding to the optimal F-score for each tool in Figure 5.2. With the best F-score of 0.9717, LncRNA-ID had the best sensitivity of 0.9660 and the FPR of 0.0225.

The optimal performance of CPAT and CPC was much better than that based on their default score cutoffs, showing that their default score cutoffs are data dependent and may not provide users

with satisfactory classification performance. The optimal performance and AUC of PhyloCSF was much worse than the other tools.

Table 5.1 Performance comparison on H1.

	LncRNA-ID	CPC	CPAT	PhyloCSF
Sensitivity	**96.73**	66.48	86.25	77.08
Specificity	95.40	**99.97**	99.42	85.08
F-score	**96.06**	79.85	92.37	80.89
Accuracy	**96.06**	83.22	92.84	81.34

CPC, CPAT and PhyloCSF were evaluated using default score cutoffs. Bold numbers indicate the highest value of the metrics.

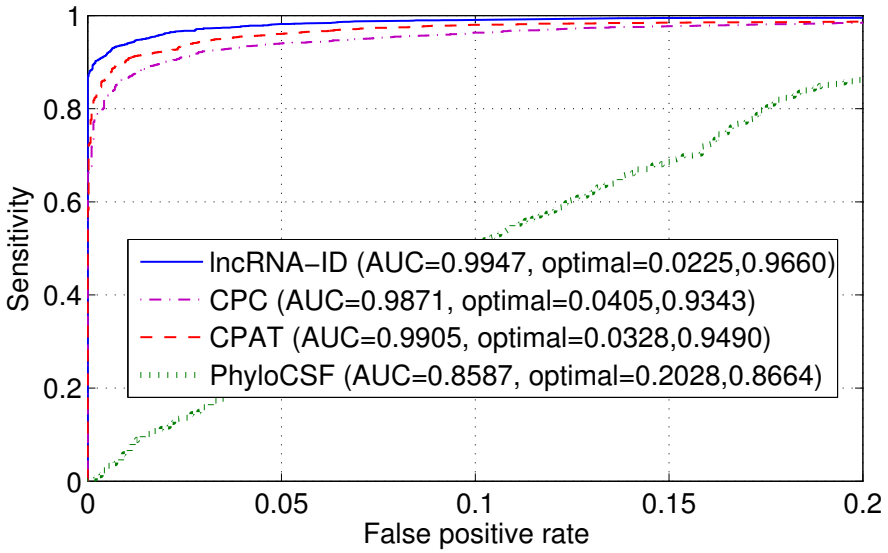


Figure 5.2 ROC curves of different tools on H1. The AUCs, and the sensitivity and FPR corresponding the optimal F-score were indicated in the legend.

5.4.3 The mouse data set (M)

LncRNA-ID can be trained for any species with some characterized protein coding and lncRNA genes. If no such training data is available, pre-built model can be used. In this experiment, we applied LncRNA-ID to the mouse data set to show its application to a different species. This data set consists of randomly selected transcripts from GENCODE. The training data contains 44,300

protein-coding transcripts and 4,000 lncRNAs. The test data contains 2,000 coding transcript and 2,000 lncRNAs. The number of lncRNAs in this data set is only half of that contained in H1 because of limited lncRNA annotation in the mouse genome.

A multiple sequence alignment of 30 genomes, including the mouse genome, was downloaded from the UCSC Genome Browser and used as the input alignment to PhyloCSF. The score cutoff of 50, which was shown to accurately separate known protein-coding genes from known non-coding sequences [130, 120], was used to generate the classification results for PhyloCSF.

Table 5.2 shows the performance comparison of different tools on the mouse data set under their default parameters. LncRNA-ID had the best sensitivity and accuracy among all tools. Although CPC and CPAT had slightly higher specificity than LncRNA-ID, their sensitivity and accuracy were much lower than those of LncRNA-ID. LncRNA-ID had the highest F-score among all tools, showing its best overall classification performance. The sensitivity, F-score, and accuracy of the tools on the mouse data set were lower than those on H1 for all except for CPC, largely due to the smaller training data set. Note that as we used CPC's pre-built classifier to evaluate the test data, there might be some overlapping samples between CPC's training data and this test data, giving an advantage to CPC's classifier over the other tools.

Figure 5.3 shows the ROC curves of different tools. When the sensitivity was lower than 0.834, CPC had lower false positive rate than the other tools. However, its optimal sensitivity was much lower than that of LncRNA-ID and CPAT. Same as on the experimental on human data set H1, LncRNA-ID had the best optimal performance and AUC. CPAT had the second highest optimal performance and AUC. At the point with the optimal F-score, CPAT's sensitivity was 2.6% lower than LncRNA-ID and its false positive rate was 0.6% lower than that of LncRNA-ID. The performance of CPAT under its default score cutoff was much lower than its optimal performance when its score cutoff was changed. CPC had slightly lower false positive rate than

LncRNA-ID when its optimal performance was achieved. However, its sensitivity was much lower than LncRNA-ID. PhyloCSF had significantly poorer performance than the other tools.

Table 5.2 Performance comparison on the mouse data set.

	LncRNA-ID	CPC	CPAT	PhyloCSF
Sensitivity	**94.65**	76.55	44.55	24.50
Specificity	92.20	**98.75**	**98.75**	55.70
F-score	**93.41**	86.24	61.40	34.02
Accuracy	**93.43**	87.65	71.65	41.43

CPC, CPAT and PhyloCSF were evaluated using default score cutoffs. Bold numbers indicate the highest value of the metrics.

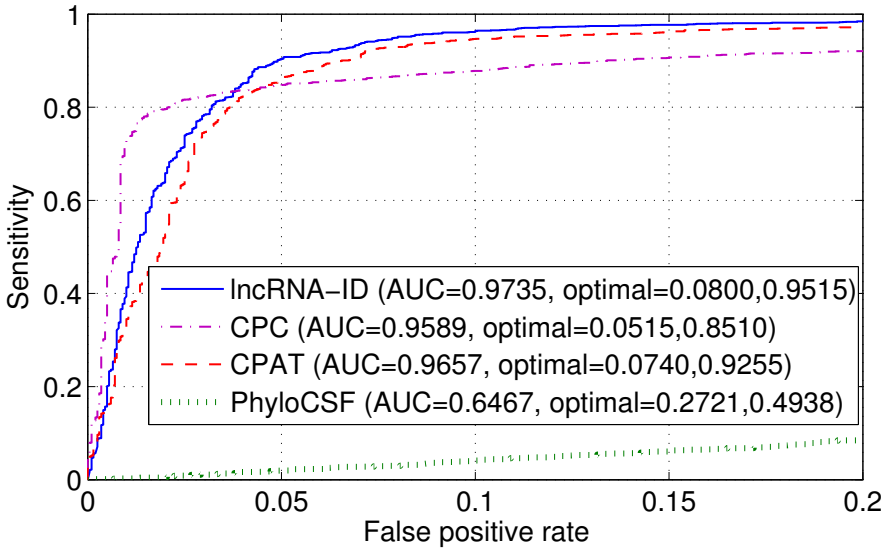


Figure 5.3 ROC curves of different tools on the mouse data set. The AUCs, and the sensitivity and FPR corresponding the optimal F-score were indicated in the legend.

5.4.4 CPAT's human data set (H2)

In this experiment, we evaluated the performance of LncRNA-ID on the human data set used by CPAT [118]. The training set was originally claimed to contain 10,000 coding transcripts selected from the RefSeq database and 10,000 randomly selected non-coding transcripts from GENCODE. However, some transcripts no longer exist in the databases. They might have been removed be-

cause of the duplication with the existing ones, non-qualification as new evidence has emerged, etc. As a result, the final training set contains 9,929 coding transcripts and 9,066 non-coding transcripts. The test set is the same as CPAT's original data. It contains 4,000 coding transcripts from RefSeq database and 4,000 lncRNAs from a human lncRNA catalog. The performance of CPC and PhyloCSF had been benchmarked on this data set in [118] and were used in our experiment. We created the classification model of CPAT from the training transcripts using the script provided in CPAT's software package. The created classifier was then used to predict the transcripts in the test data set and the performance was evaluated using the CPAT's suggested optimal score cutoff.

The first three columns of Table 5.3 shows the performance comparison between LncRNA-ID and CPC. LncRNA-ID (sensitivity: 93.65%, specificity: 96.15%, F-score: 94.88%) achieved a better overall performance compared with CPAT (sensitivity: 87.58%, specificity: 97.32%, F-score: 92.20%), CPC (sensitivity: 73.75%, specificity: 99.9%, F-score: 84.85%), and PhyloCSF (F-score: 74.05%, sensitivity: 62.8%, specificity: 90.2%). Please note that we used the performances of CPC and PhyloCSF which were benchmarked on the same test data in [118].

Table 5.3 Performance comparison on H2

	LncRNA-ID	CPAT	LncRNA-ID				CPAT			
	Original data		S2	S3	S6	S8	S2	S3	S6	S8
Sensitivity	**93.65**	87.58	93.43	93.54	92.72	92.73	79.51	73.01	60.32	54.46
Specificity	96.15	**97.32**	95.41	95.23	95.47	95.38	98.14	98.50	99.29	99.42
F-score	**94.88**	92.20	94.41	94.38	94.08	94.03	87.84	83.85	75.03	70.32
Accuracy	**94.90**	92.45	94.42	94.38	94.10	94.06	88.82	85.75	79.81	76.94

Bold numbers indicate the highest values of the metrics.

5.4.5 Imbalanced training data

Using H2 data set, we evaluated how imbalanced training data affects the classification performance of LncRNA-ID and CPAT, which are the two best tools according to previous experimental

results. We constructed four data sets, S2, S3, S6, and S8, from the original training set simulating the condition of imbalanced training data. In S2, lncRNAs in the training set were randomly divided into two subsets. Each lncRNA subset was combined with coding transcripts in the original training set, generating two training subsets in total. We created the classification models of LncRNA-ID and CPAT using each of the two training subsets and evaluated their performance using the same test set. The experiments on S3, S6, and S8 were conducted in the same manner except that lncRNAs were divided into three, six, and eight subsets, respectively.

LncRNA-ID had the higher average sensitivity than CPAT in all four experiments (Table 5.3). The performance of CPAT's classifiers trained with the subsets significantly decreased compared with that trained with the full training set. This shows that limited learning data led to less discriminative power of CPAT's classifier. In contrast, LncRNA-ID, which implements balanced random forest learning, was able to maintain stable performance in all data sets with different ratios of imbalance.

The average specificity of LncRNA-ID was 3.46% lower than CPAT. However, the average sensitivity of CPAT was 26.28% lower than that of LncRNA-ID. The average accuracy and F-score of LncRNA-ID were also much higher than those of CPC, showing better overall classification performance. CPAT's sensitivities dramatically dropped by 9.21-37.81% while LncRNA-ID's sensitivity decreased by less than 1% compared with those trained with the full training set. This shows that CPAT suffered not only from the limited learning data but also the impact of imbalanced training data.

5.4.6 Running time

We measured the running time of LncRNA-ID compared to CPC, CPAT, and PhyloCSF on the H1 test set, which is the largest test set. All tools were ran on the same high performance computing

node that has 64 bits CPU with Linux operating system. CPC, CPAT, and PhyloCSF took 86.51h, 35.36s and 15,097.60h to process the data. Note that the running time of PhyloCSF did not include the time used for preparing the input multiple sequence alignments, which can be computationally expensive. LncRNA-ID took 65.36s to process the data. Its speed was comparable to CPAT, and much faster than CPC and PhyloCSF.

5.5 Discussion and conclusion

We have proposed LncRNA-ID, an accurate lncRNA identification using balanced random forest classification. LncRNA-ID ensembles multiple forest classifiers induced from balanced down-sampled data and thus is able to maintain steady performance with different ratios of imbalance and limited learning data. The results in both human and mouse genome demonstrates that the features used by LncRNA-ID have powerful discriminative power in distinguishing lncRNAs from protein-coding transcripts. Our empirical experiment shows that the ribosome interaction features are the most discriminating features.

Among all classification tools, PhyloCSF had the worst performance. The explanation of this result is that in the human data set, PhyloCSF could not determine the coding status of a decent amount of lncRNAs (16.97%) and some coding transcripts (0.03%). This is because they either are non-conserved transcripts or do not have sufficient long ORFs.

If ribosome profiling data (Ribo-Seq) and mRNA-Seq data are available, a more accurate ribosome release signal could be measured using the numbers of mapped reads on ORF and 3'UTR. The RRS is then defined as the ratio of the two normalized ratios of mapped reads on these two regions, $RRS = (r_{ORF}/r_{3'UTR})_{Ribo-Seq}/(r_{ORF}/r_{3'UTR})_{mRNA-Seq}$ [129], where r is a number of mapped reads on a sequence.

LncRNA-ID achieved the highest overall performances on both human and mouse compared with other tools. The performance of LncRNA-ID is even more pronounced when learning with imbalanced data set. The imbalanced learning data is essentially found in most species, in which there are a large amount of functional annotation of proteins while validated annotation of lncRNAs are far less. The ability to maintain steady performance of LncRNA-ID on limited and imbalanced data results from applying balanced random forest learning. LncRNA-ID employs the down-sampling technique to create multiple balanced learning data sets from the original imbalanced data. A balanced learning data prevents a classifier from being biased to the majority class. Moreover, LncRNA-ID uses all learning data in classification by integrating down-sampled data into multiple classifiers and thus prevents loss of information.

Chapter 6

Conclusion and future work

Next Generation Sequencing (NGS) technologies have greatly extended the abilities of researchers to intensively study gene expression and discover new coding and non-coding genes on a genome-wide scale. NcNRAs have gained significant increasing interest as many evidences have revealed their critical roles in various biological processes. Nevertheless, NGS has identified many transcripts whose functions and significance are unclear [152]. In support of the growing attention in ncRNAs identification, we have proposed four tools: grammar-string based alignment, KnotShape, KnotStructure, and LncRNA-ID.

We proposed the grammar-string, a novel secondary structure representation and showed its application in consensus structure derivation through multiple ncRNA alignment. Compared with existing structure prediction tools, it had better sensitivity with high positive predictive value. We also showed the utility of grammar string alignment in consensus structure derivation for ncRNAs including pseudoknots. Besides constructing ncRNA structural alignment, grammar string can be used to encode characterized ncRNA structures (such as those from Rfam), compare different structures, and search for common structural motifs. We plan to explore these utilities of grammar strings.

KnotShape and KnotStructure were designed specifically to decrease the searching space of putative structures of homologous RNA sequences. The combination of mass data analysis and SVM-based feature ranking makes KnotShape a promising tool for shape prediction. By combin-

ing the predicted shapes and the multiple structural alignment strategy, KnotStructure demonstrates higher accuracy in pseudoknot structure prediction. There are some improvements that could be made to further increase the sensitivity of the shape ranking step. In shape ranking, there were a few input sequence sets for which the energies of correct structures are not near-optimal. Thus, enlarging the sample folding space will likely increase the sensitivity. However, using a large number of sub-optimal structures can increase the computational cost. Thus, a better algorithm is needed to achieve a better tradeoff between sensitivity and running time.

LncRNA-ID was specifically designed to identify lncRNAs, which are different from other ncRNAs in that they are: i)longer than 200 nucleotides; ii) lack of strong sequence conservation across species; iii) usually have low to medium expression levels with no specific pattern; iv) found to have ORFs as long as those found in protein-coding transcripts. LncRNA-ID is the lncRNA classification using balanced random forest based on eleven biological coherent features. LncRNA-ID focused on distinguishing lncRNAs from protein-coding transcript, which is the first critical step for understanding the underneath biological roles. LncRNA-ID ensembles multiple forest classifiers induced from balanced down-sampled data and thus is able to maintain the steady performance with different imbalance ratio and limited learning data. In our experiments, we focused on the ribosome interaction in eukaryotes where more annotated lncRNAs are publicly available. Although LncRNA-ID showed great classification performance in two important eukaryotes, human and mouse, it has not been tested on prokaryotes. The difference between eukaryotes and prokaryotes is that in eukaryotes, a small 40S ribosomal subunit contains 18S rRNA whereas in prokaryotes a small 30S ribosomal subunit contains 16S rRNA. Thus, further study is needed to investigate whether the similar ribosome signal could be captured with 16S rRNA as with 18S rRNA. The running time of protein-conservation feature extraction is the bottleneck of lncRNAs, especially for large numbers of transcripts. As we used the profile HMM-based methods to reflect

the protein-conservation of transcripts, this can be naturally parallelized. This methodology has greatly improved the scalability of the analysis of large-scale data.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker. From sequences to shapes and back: a case study in RNA secondary structures.*Proceedings of the Royal Society: Biological Sciences, Series B*, 255(1344):279–284, 1994.

[2] E. Pennisi. Genomics. ENCODE project writes eulogy for junk DNA.*Science*, 337(6099):1159, 1161, Sep 2012.

[3] P. Kapranov, J. Cheng, S. Dike, D. A. Nix, R. Duttagupta, A. T. Willingham, P. F. Stadler, J. Hertel, J. Hackermuller, I. L. Hofacker, I. Bell, E. Cheung, J. Drenkow, E. Dumais, S. Patel, G. Helt, M. Ganesh, S. Ghosh, A. Piccolboni, V. Sementchenko, H. Tammana, and T. R. Gingeras. RNA maps reveal new RNA classes and a possible function for pervasive transcription.*Science*, 316(5830):1484–1488, Jun 2007.

[4] S. Djebali, C. A. Davis, A. Merkel, A. Dobin, T. Lassmann, A. Mortazavi, A. Tanzer, J. Lagarde, W. Lin, F. Schlesinger, C. Xue, G. K. Marinov, J. Khatun, B. A. Williams, C. Zaleski, J. Rozowsky, M. Roder, F. Kokocinski, R. F. Abdelhamid, T. Alioto, I. Antoshechkin, M. T. Baer, N. S. Bar, P. Batut, K. Bell, I. Bell, S. Chakrabortty, X. Chen, J. Chrast, J. Curado, T. Derrien, J. Drenkow, E. Dumais, J. Dumais, R. Duttagupta, E. Falconnet, M. Fastuca, K. Fejes-Toth, P. Ferreira, S. Foissac, M. J. Fullwood, H. Gao, D. Gonzalez, A. Gordon, H. Gunawardena, C. Howald, S. Jha, R. Johnson, P. Kapranov, B. King, C. Kingswood, O. J. Luo, E. Park, K. Persaud, J. B. Preall, P. Ribeca, B. Risk, D. Robyr, M. Sammeth, L. Schaffer, L. H. See, A. Shahab, J. Skancke, A. M. Suzuki, H. Takahashi, H. Tilgner, D. Trout, N. Walters, H. Wang, J. Wrobel, Y. Yu, X. Ruan, Y. Hayashizaki, J. Harrow, M. Gerstein, T. Hubbard, A. Reymond, S. E. Antonarakis, G. Hannon, M. C. Giddings, Y. Ruan, B. Wold, P. Carninci, R. Guigo, and T. R. Gingeras. Landscape of transcription in human cells.*Nature*, 489(7414):101–108, Sep 2012.

[5] J. E. Wilusz, H. Sunwoo, and D. L. Spector. Long noncoding RNAs: functional surprises from the RNA world.*Genes Dev.*, 23(13):1494–1504, Jul 2009.

[6] A. Pauli, J. L. Rinn, and A. F. Schier. Non-coding RNAs as regulators of embryogenesis. *Nat. Rev. Genet.*, 12(2):136–149, Feb 2011.

[7] D. P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function.*Cell*, 116(2):281–297, Jan 2004.

[8] Matthew W. W. Jones-Rhoades, David P. P. Bartel, and Bonnie Bartel. MicroRNAs and their regulatory roles in plants.*Annual Review of Plant Biology*, 57:19–53, 2006.

[9] T. Hung and H. Y. Chang. Long noncoding RNA in genome regulation: prospects and mechanisms.*RNA Biol*, 7(5):582–585, 2010.

[10] T. Derrien, R. Johnson, G. Bussotti, A. Tanzer, S. Djebali, H. Tilgner, G. Guernec, D. Martin, A. Merkel, D. G. Knowles, J. Lagarde, L. Veeravalli, X. Ruan, Y. Ruan, T. Lassmann, P. Carninci, J. B. Brown, L. Lipovich, J. M. Gonzalez, M. Thomas, C. A. Davis, R. Shiekhattar, T. R. Gingeras, T. J. Hubbard, C. Notredame, J. Harrow, and R. Guigo. The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression.*Genome Res.*, 22(9):1775–1789, Sep 2012.

[11] D. Bu, K. Yu, S. Sun, C. Xie, G. Skogerb, R. Miao, H. Xiao, Q. Liao, H. Luo, G. Zhao, H. Zhao, Z. Liu, C. Liu, R. Chen, and Y. Zhao. NONCODE v3.0: integrative annotation of long noncoding RNAs.*Nucleic Acids Res.*, 40(Database issue):D210–215, Jan 2012.

[12] Disease-related human lncRNA Profiler. http://www.systembio.com/lncrna-research/disease-long-non-coding-rna.

[13] O. Wapinski and H. Y. Chang. Long noncoding RNAs and human disease.*Trends Cell Biol.*, 21(6):354–361, 2011.

[14] Sean R Eddy. Computational genomics of noncoding rna genes.*Cell*, 109(2):137–140, April 2002.

[15] Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information.*Nucl. Acids Res.*, 9(1):133–148, 1981.

[16] D.H. Mathews, M.D. Disney, J. L. Childs, S.J. Schroeder, M. Zuker, and D.H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure.*Proceedings of the National Academy of Sciences USA*, 101:7287–7292, 2004.

[17] D.H. Mathews. Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization.*RNA*, 10:1178–1190, 2004.

[18] J. S. Mccaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure.*Biopolymers*, 29(6-7):1105–1119, 1990.

[19] M. Andronescu, C. Pop, and A. Condon. Improved energy parameters for RNA pseudoknotted secondary structure prediction.*RNA*, 16(1):26–42, 2010.

[20] D K Treiber and J R Williamson. Exposing the kinetic traps in RNA folding.*Curr Opin Struct Biol*, 9(3):339–45, 1999.

[21] A. Adams, T. Lindahl, and J. R. Fresco. Conformational differences between the biologically active and inactive forms of a transfer ribonucleic acid.*Proc. Natl. Acad. Sci. U.S.A.*, 57(6):1684–1691, Jun 1967.

[22] S. Washietl, I. L. Hofacker, and P. F. Stadler. Fast and reliable prediction of noncoding RNAs.*Proc Natl Acad Sci U S A*, 102(7):2454–2459, 2005.

[23] R. B. Lyngsøand C. N. Pedersen. RNA pseudoknot prediction in energy-based models. *Journal of Computational Biology*, 7(3–4):409–427, 2000.

[24] J. N. Hutchinson, A. W. Ensminger, C. M. Clemson, C. R. Lynch, J. B. Lawrence, and A. Chess. A screen for nuclear transcripts identifies two linked noncoding RNAs associated with SC35 splicing domains.*BMC Genomics*, 8:39, 2007.

[25] H. Sunwoo, M. E. Dinger, J. E. Wilusz, P. P. Amaral, J. S. Mattick, and D. L. Spector. MEN epsilon/beta nuclear-retained non-coding RNAs are up-regulated upon muscle differentiation and are essential components of paraspeckles.*Genome Res.*, 19(3):347–359, Mar 2009.

[26] L. L. Chen and G. G. Carmichael. Altered nuclear retention of mRNAs containing inverted repeats in human embryonic stem cells: functional role of a nuclear noncoding RNA.*Mol. Cell*, 35(4):467–478, Aug 2009.

[27] Z. Wang, M. Gerstein, and M. Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, 10(1):57–63, Jan 2009.

[28] B. Banfai, H. Jia, J. Khatun, E. Wood, B. Risk, W. E. Gundling, A. Kundaje, H. P. Gunawardena, Y. Yu, L. Xie, K. Krajewski, B. D. Strahl, X. Chen, P. Bickel, M. C. Giddings, J. B. Brown, and L. Lipovich. Long noncoding RNAs are rarely translated in two human cell lines.*Genome Res.*, 22(9):1646–1657, Sep 2012.

[29] Ingolia NT Weissman JS Guttman M, Russell P and Lander ES. Ribosome Profiling Provides Evidence that Large Noncoding RNAs Do Not Encode Proteins.*Cell*, 154:240–251, 2013.

[30] K. V. Prasanth and D. L. Spector. Eukaryotic regulatory RNAs: an answer to the 'genome complexity' conundrum.*Genes Dev.*, 21(1):11–42, Jan 2007.

[31] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison.*Biological Sequence Analysis Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, UK, 1998.

[32] Ivo L. Hofacker, Stephan H. F. Bernhart, and Peter F. Stadler. Alignment of RNA base pairing probability matrices.*Bioinformatics*, 20(14):2222–2227, 2004.

[33] Sebastian Will, Kristin Reiche, Ivo L Hofacker, Peter F Stadler, and Rolf Backofen. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering.*PLoS Comput Biol*, 3(4):e65, 2007.

[34] Elfar Torarinsson, Jakob H. Havgaard, and Jan Gorodkin. Multiple structural alignment and clustering of RNA sequences.*Bioinformatics*, 23(8):926–932, 2007.

[35] Matthias Höchsmann, Thomas Töller, Robert Giegerich, and Stefan Kurtz. Local similarity in RNA secondary structures. In*CSB '03: Proceedings of the IEEE Computer Society Conference on Bioinformatics*, page 159, Washington, DC, USA, 2003. IEEE Computer Society.

[36] M. Höchsmann, B. Voss, and R. Giegerich. Pure multiple RNA secondary structure alignments: a progressive profile approach.*IEEE/ACM Trans Comput Biol Bioinform*, 1(1):53–62, 2004.

[37] Helene Touzet and Olivier Perriquet. CARNAC: folding families of related RNAs.*Nucl. Acids Res.*, 32(suppl. 2):W142–145, 2004.

[38] S Karlin and S F Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes.*Proc Natl Acad Sci*, 87(6):2264–2268, 1990.

[39] Comparing RNA secondary structures based on 2d graphical representation.

[40] F Bai, D Li, and T Wang. A new mapping rule for RNA secondary structures with its applications.*J. Math. Chem.*, 43:932–943, 2008.

[41] C Li, A H Wang, and L Xing. Similarity of RNA secondary structures.*J. Comput. Chem.*, 28:508–512, 2007.

[42] Kishore Doshi, Jamie Cannone, Christian Cobaugh, and Robin Gutell. Evaluation of the suitability of free-energy minimization using nearest-neighbour energy parameters for RNA secondary structure prediction.*BMC Bioinformatics*, 5(1):105, 2004.

[43] David Sankoff. Simultaneous Solution of the RNA Folding, Alignment and Protosequence Problems.*SIAM Journal on Applied Mathematics*, 45(5):810–825, 1985.

[44] Ian Holmes. A probabilistic model for the evolution of rna structure.*BMC Bioinformatics*, 5(1):166, 2004.

[45] Robin D Dowell and Sean R Eddy. Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints.*BMC Bioinformatics*, 7(400), 2006.

[46] Sven Siebert and Rolf Backofen. MARNA: multiple alignment and consensus structure prediction of RNAs based on sequence structure comparisons.*Bioinformatics*, 21(16):3352–3359, 2005.

[47] IL Hofacker, M Fekete, and PF Stadler. Secondary structure prediction for aligned RNA sequences.*J Mol Biol.*, 319(5):1059–66, 2002.

[48] Elena Rivas and Sean R. Eddy. Noncoding RNA gene detection using comparative sequence analysis.*BMC Bioinformatics*, 2(1):8, 2001.

[49] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars.*Nucleic Acids Res*, 31(13):3423–3428, 2003.

[50] M. Blanchette, W. J. Kent, C. Riemer, L. Elnitski, A. F. Smit, K. M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E. D. Green, D. Haussler, and W. Miller. Aligning multiple genomic sequences with the threaded blockset aligner.*Genome Res*, 14(4):708–715, 2004.

[51] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.*Nucl. Acids Res.*, 22(22):4673–4680, 1994.

[52] I.L. Hofacker. Vienna RNA secondary structure server.*Nucleic Acids Res.*, 31:3429–3431, 2003.

[53] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In*NIPS*, pages 487–493, 1998.

[54] Robin Dowell and Sean Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction.*BMC Bioinformatics*, 5(1):71, 2004.

[55] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool.*Journal of Molecular Biology*, 215:403–410, 1990.

[56] Needleman SB and Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins.*J Mol Biol*, 48:443–453, 1970.

[57] S. Griffiths-Jones, S. Moxon, M. Marshall, A. Khanna, S. R. Eddy, and A. Bateman. Rfam: annotating non-coding RNAs in complete genomes.*Nucleic Acids Res.*, 33:D121–D124, 2005.

[58] Chuong B. Do, Daniel A. Woods, and Serafim Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models.*Bioinformatics*, 22(14):e90–98, 2006.

[59] N. R. Markham and M. Zuker. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Res.*, 33:W577–W581, 2005.

[60] N. R. Markham and M. Zuker. UNAFold: software for nucleic acid folding and hybrization. In*Bioinformatics, Volume II. Structure, Functions and Applications, number 453 in Methods in Molecular Biology*, pages 3–31, Totowa, NJ, USA, 2008. Humana Press.

[61] D.F. Feng and R.F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees.*J Mol Evol.*, 25(4):351–360, 1987.

[62] Hisanori Kiryu, Yasuo Tabei, Taishin Kin, and Kiyoshi Asai. Murlet: a practical multiple alignment tool for structural RNA sequences.*Bioinformatics*, 23(13):1588–1598, 2007.

[63] A. Wilm, I Mainz, and G. Steger. An enhanced RNA alignment benchmark for sequence alignment programs.*Algorithms Mol Biol.*, 1(19), 2006.

[64] D. W. Staple and S. E. Butcher. Pseudoknots: RNA structures with diverse functions.*PLoS Biol.*, 3(6):e213, 2005.

[65] E. Rivas and S. Eddy. The language of RNA: a formal grammar that includes pseudoknots. *Bioinformatics*, 16(4):334–340, 2000.

[66] X. Xu, Y. Ji, and G. D. Stormo. RNA Sampler: a new sampling based algorithm for common RNA secondary structure prediction and structural alignment.*Bioinformatics*, 23(15):1883–1891, 2007.

[67] J. Ruan, G. D. Stormo, and W. Zhang. ILM: a web server for predicting RNA secondary structures with pseudoknots.*Nucleic Acids Res.*, 32(suppl 2):W146–W149, 2004.

[68] Paul P Gardner, Jennifer Daub, John Tate, Benjamin L Moore, Isabelle H Osuch, Sam Griffiths-Jones, Robert D Finn, Eric P Nawrocki, Diana L Kolbe, Sean R Eddy, and et al. Rfam: Wikipedia, clans and the “decimal" release.*Nucleic Acids Research*, 39:D141–D145, 2011.

[69] Paul Gardner and Robert Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches.*BMC Bioinformatics*, 5(1):140, 2004.

[70] Stephan H. Bernhart and Ivo L. Hofacker. From consensus structure prediction to RNA gene finding.*Briefings in functional genomics & proteomics*, 8(6):461–471, 2009.

[71] Robert Giegerich, Björn Voß, and Marc Rehmsmeier. Abstract shapes of RNA.*Nucleic Acids Research*, 32(16):4843–4851, 2004.

[72] Anika Scheid and Markus Nebel. On abstract shapes of RNA. Technical Report 368, Informatik, 2008.

[73] Jens Reeder and Robert Giegerich. Consensus shapes: an alternative to the sankoff algorithm for RNA consensus structure prediction.*Bioinformatics*, 21(17):3516–3523, 2005.

[74] Eugene Berezikov, Geert van Tetering, Mark Verheul, Jose van de Belt, Linda van Laake, Joost Vos, Robert Verloop, Marc van de Wetering, Victor Guryev, Shuji Takada, Anton Jan van Zonneveld, Hiroyuki Mano, Ronald Plasterk, and Edwin Cuppen. Many novel mammalian microRNA candidates identified by extensive cloning and RAKE analysis.*Genome Res*, 16(10):1289–98, 2006.

[75] Jian Lu, Yang Shen, Qingfa Wu, Supriya Kumar, Bin He, Suhua Shi, Richard W Carthew, San Ming Wang, and Chung-I Wu. The birth and death of microRNA genes in Drosophila. *Nat Genet*, 40(3):351–5, 2008.

[76] S. Janssen, J. Reeder, and R. Giegerich. Shape based indexing for faster search of rna family databases.*BMC Bioinformatics*, 9(1):131, 2008.

[77] Chun-Hsiang Huang, Chin Lung Lu, and Hsien-Tai Chiu. A heuristic approach for detecting RNA H-type pseudoknots.*Bioinformatics*, 21(17):3501–3508, 2005.

[78] Song Cao and Shi-Jie Chen. Predicting structures and stabilities for H-type pseudoknots with interhelix loops.*RNA*, 15:696–706, 2009.

[79] Jana Sperschneider and Amitava Datta. DotKnot: pseudoknot prediction using the probability dot plot under a refined energy model.*Nucleic Acids Research*, 38(7):e103, 2010.

[80] Kengo Sato, Yuki Kato, Michiaki Hamada, Tatsuya Akutsu, and Kiyoshi Asai. IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming.*Bioinformatics*, 27(13):i85–i93, 2011.

[81] Michael Bon and Henri Orland. TT2NE: a novel algorithm to predict RNA secondary structures with pseudoknots.*Nucleic Acids Research*, 39(14):e93, 2011.

[82] Matthew G. Seetin and David H Mathews. TurboKnot: rapid prediction of conserved RNA secondary structures including pseudoknots.*Bioinformatics*, 28(6):792–798, 2012.

[83] Stefan Janssen and Robert Giegerich. Faster computation of exact RNA shape probabilities. *Bioinformatics*, 26(5):632–639, 2010.

[84] Peter Steffen, Björn Voß, Marc Rehmsmeier, Jens Reeder, and Robert Giegerich. RNAshapes: an integrated RNA analysis package based on abstract shapes.*Bioinformatics*, 22(4):500–503, 2006.

[85] Björn Voss, Robert Giegerich, and Marc Rehmsmeier. Complete probabilistic analysis of RNA shapes.*BMC biology*, 4(1):5, 2006.

[86] Samuel S. Cho, David L. Pincus, and D. Thirumalai. Assembly mechanisms of RNA pseudoknots are determined by the stabilities of constituent secondary structures.*Proceedings of the National Academy of Sciences*, 106(41):17349–17354, 2009.

[87] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In*Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, USA, 1992.

[88] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure.*Journal of molecular biology*, 288(5):911–940, 1999.

[89] Yi-wei Chen. Combining SVMs with various feature selection strategies. In*Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag, 2005.

[90] K Z Mao. Fast orthogonal forward selection algorithm for feature subset selection.*IEEE Transactions on Neural Networks*, 13(5):1218–1224, 2002.

[91] T. Marill and D. Green. On the effectiveness of receptors in recognition systems.*Information Theory, IEEE Transactions on*, 9(1):11–17, 1963.

[92] Tong Zhang.*Fundamental Statistical Techniques, Chapter in Handbook of Natural Language Processing*. Chapman and Hall, 2010.

[93] Stephen Johnson. Hierarchical clustering schemes.*Psychometrika*, 32(3):241–254, 1967.

[94] Rujira Achawanantakun, Seyedeh Shohreh Takyar, and Yanni Sun. Grammar string: a novel ncRNA secondary structure representation.*lifesciences society org*, pages 2–13, 2010.

[95] Rujira Achawanantakun, Yanni Sun, and Seyedeh Shohreh Takyar. ncRNA consensus secondary structure derivation using grammar strings.*J. Bioinformatics and Computational Biology*, 9(2):317–337, 2011.

[96] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[97] Christina Witwer, Ivo L Hofacker, and Peter F Stadler. Prediction of consensus RNA secondary structures including pseudoknots.*IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(2):66–77, 2004.

[98] Jihong Ren, Baharak Rastegari, Anne Condon, and Holger H Hoos. HotKnots: heuristic prediction of RNA secondary structures including pseudoknots.*RNA*, 11:1494–1504, 2005.

[99] I. Bentwich, A. Avniel, Y. Karov, R. Aharonov, S. Gilad, O. Barad, A. Barzilai, P. Einat, U. Einav, E. Meiri, E. Sharon, Y. Spector, and Z. Bentwich. Identification of hundreds of conserved and nonconserved human microRNAs.*Nat. Genet.*, 37(7):766–770, Jul 2005.

[100] M. A. Blasco. Telomere length, stem cells and aging.*Nat. Chem. Biol.*, 3(10):640–649, Oct 2007.

[101] P. Carninci and et al. The transcriptional landscape of the mammalian genome.*Science*, 309(5740):1559–1563, Sep 2005.

[102] J. Liu, C. Jung, J. Xu, H. Wang, S. Deng, L. Bernad, C. Arenas-Huertero, and N. H. Chua. Genome-wide analysis uncovers regulation of long intergenic noncoding RNAs in Arabidopsis.*Plant Cell*, 24(11):4333–4345, Nov 2012.

[103] S. Boerner and K. M. McGinnis. Computational identification and functional predictions of long noncoding RNA in Zea mays.*PLoS ONE*, 7(8):e43047, 2012.

[104] F. C. Humann, G. J. Tiberio, and K. Hartfelder. Sequence and Expression Characteristics of Long Noncoding RNAs in Honey Bee Caste Development - Potential Novel Regulators for Transgressive Ovary Size.*PLoS ONE*, 8(10):e78915, 2013.

[105] C. Arriaga-Canon, Y. Fonseca-Guzman, C. Valdes-Quezada, R. Arzate-Mejia, G. Guerrero, and F. Recillas-Targa. A long non-coding RNA promotes full activation of adult gene expression in the chicken globin domain.*Epigenetics*, 9(1), Nov 2013.

[106] Y. Liu, D. Luo, H. Zhao, Z. Zhu, W. Hu, and C. H. Cheng. Inheritable and Precise Large Genomic Deletions of Non-Coding RNA Genes in Zebrafish Using TALENs.*PLoS ONE*, 8(10):e76387, 2013.

[107] G. Chen, Z. Wang, D. Wang, C. Qiu, M. Liu, X. Chen, Q. Zhang, G. Yan, and Q. Cui. LncRNADisease: a database for long-non-coding RNA-associated diseases.*Nucleic Acids Res.*, 41(Database issue):D983–986, Jan 2013.

[108] Y. Okazaki, M. Furuno, T. Kasukawa, J. Adachi, H. Bono, S. Kondo, I. Nikaido, N. Osato, R. Saito, H. Suzuki, I. Yamanaka, H. Kiyosawa, K. Yagi, Y. Tomaru, Y. Hasegawa, A. Nogami, C. Schonbach, T. Gojobori, R. Baldarelli, D. P. Hill, C. Bult, D. A. Hume, J. Quackenbush, L. M. Schriml, A. Kanapin, H. Matsuda, S. Batalov, K. W. Beisel, J. A. Blake, D. Bradt, V. Brusic, C. Chothia, L. E. Corbani, S. Cousins, E. Dalla, T. A. Dragani, C. F. Fletcher, A. Forrest, K. S. Frazer, T. Gaasterland, M. Gariboldi, C. Gissi, A. Godzik, J. Gough, S. Grimmond, S. Gustincich, N. Hirokawa, I. J. Jackson, E. D. Jarvis, A. Kanai, H. Kawaji, Y. Kawasawa, R. M. Kedzierski, B. L. King, A. Konagaya, I. V. Kurochkin, Y. Lee, B. Lenhard, P. A. Lyons, D. R. Maglott, L. Maltais, L. Marchionni, L. McKenzie, H. Miki, T. Nagashima, K. Numata, T. Okido, W. J. Pavan, G. Pertea, G. Pesole, N. Petrovsky, R. Pillai, J. U. Pontius, D. Qi, S. Ramachandran, T. Ravasi, J. C. Reed, D. J. Reed, J. Reid, B. Z. Ring, M. Ringwald, A. Sandelin, C. Schneider, C. A. Semple, M. Setou, K. Shimada, R. Sultana, Y. Takenaka, M. S. Taylor, R. D. Teasdale, M. Tomita, R. Verardo, L. Wagner, C. Wahlestedt, Y. Wang, Y. Watanabe, C. Wells, L. G. Wilming, A. Wynshaw-Boris, M. Yanagisawa, I. Yang, L. Yang, Z. Yuan, M. Zavolan, Y. Zhu, A. Zimmer, P. Carninci, N. Hayatsu, T. Hirozane-Kishikawa, H. Konno, M. Nakamura, N. Sakazume, K. Sato, T. Shiraki, K. Waki, J. Kawai, K. Aizawa, T. Arakawa, S. Fukuda, A. Hara, W. Hashizume, K. Imotani, Y. Ishii, M. Itoh, I. Kagawa, A. Miyazaki, K. Sakai, D. Sasaki, K. Shibata, A. Shinagawa, A. Yasunishi, M. Yoshino, R. Waterston, E. S. Lander, J. Rogers, E. Birney, and Y. Hayashizaki. Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cDNAs.*Nature*, 420(6915):563–573, Dec 2002.

[109] M. E. Dinger, K. C. Pang, T. R. Mercer, and J. S. Mattick. Differentiating protein-coding and noncoding RNA: challenges and ambiguities.*PLoS Comput. Biol.*, 4(11):e1000176, Nov 2008.

[110] N. Brockdorff, A. Ashworth, G. F. Kay, V. M. McCabe, D. P. Norris, P. J. Cooper, S. Swift, and S. Rastan. The product of the mouse Xist gene is a 15 kb inactive X-specific transcript containing no conserved ORF and located in the nucleus.*Cell*, 71(3):515–526, Oct 1992.

[111] G. Borsani, R. Tonlorenzi, M. C. Simmler, L. Dandolo, D. Arnaud, V. Capra, M. Grompe, A. Pizzuti, D. Muzny, C. Lawrence, H. F. Willard, P. Avner, and A. Ballabio. Characterization of a murine gene expressed from the inactive X chromosome.*Nature*, 351(6324):325–329, May 1991.

[112] L. Kong, Y. Zhang, Z. Q. Ye, X. Q. Liu, S. Q. Zhao, L. Wei, and G. Gao. CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Res.*, 35(Web Server issue):W345–349, Jul 2007.

[113] M. F. Lin, I. Jungreis, and M. Kellis. PhyloCSF: a comparative genomics method to distinguish protein coding and non-coding regions.*Bioinformatics*, 27(13):i275–282, Jul 2011.

[114] A. Marchler-Bauer, C. Zheng, F. Chitsaz, M. K. Derbyshire, L. Y. Geer, R. C. Geer, N. R. Gonzales, M. Gwadz, D. I. Hurwitz, C. J. Lanczycki, F. Lu, S. Lu, G. H. Marchler, J. S. Song, N. Thanki, R. A. Yamashita, D. Zhang, and S. H. Bryant. CDD: conserved domains and protein three-dimensional structure.*Nucleic Acids Res.*, 41(Database issue):D348–352, Jan 2013.

[115] A. Marchler-Bauer and S. H. Bryant. CD-Search: protein domain annotations on the fly. *Nucleic Acids Res.*, 32(Web Server issue):W327–331, Jul 2004.

[116] R. A. Chodroff, L. Goodstadt, T. M. Sirey, P. L. Oliver, K. E. Davies, E. D. Green, Z. Molnar, and C. P. Ponting. Long noncoding RNA genes: conservation of sequence and brain expression among diverse amniotes.*Genome Biol.*, 11(7):R72, 2010.

[117] Patrick D. Schloss. The effects of alignment quality, distance calculation method, sequence filtering, and region on the analysis of 16s rrna gene-based studies.*PLoS Computational Biology*, 6(7), 2010.

[118] L. Wang, H. J. Park, S. Dasari, S. Wang, J. P. Kocher, and W. Li. CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model.*Nucleic Acids Res.*, 41(6):e74, Apr 2013.

[119] Foster Probost. Machine learning from imbalanced data sets 101, 2000.

[120] M. Guttman, M. Garber, J. Z. Levin, J. Donaghey, J. Robinson, X. Adiconis, L. Fan, M. J. Koziol, A. Gnirke, C. Nusbaum, J. L. Rinn, E. S. Lander, and A. Regev. Chromatin sig-

nature reveals over a thousand highly conserved large non-coding RNAs.*Nat. Biotechnol.*, 458:223–227, February 2009.

[121] Andrea Pauli, Eivind Valen, Michael F. Lin, Manuel Garber, Nadine L. Vastenhouw, Joshua Z. Levin, Lin Fan, Albin Sandelin, John L. Rinn, Aviv Regev, and Alexander F. Schier. Systematic identification of long noncoding rnas expressed during zebrafish embryogenesis. 22:577–591+, 2012.

[122] Leo Breiman. Random forests.*Machine Learning*, 45(1):5–32, 2001.

[123] J. Wu, H. Liu, X. Duan, Y. Ding, H. Wu, Y. Bai, and X. Sun. Prediction of DNA-binding residues in proteins from amino acid sequences using a random forest model with a hybrid feature.*Bioinformatics*, 25(1):30–35, Jan 2009.

[124] Y. Y. Leung, P. Ryvkin, L. H. Ungar, B. D. Gregory, and L. S. Wang. CoRAL: predicting non-coding RNAs from small RNA-sequencing data.*Nucleic Acids Res.*, 41(14):e137, Aug 2013.

[125] X. Chen and H. Ishwaran. Random forests for genomic data analysis.*Genomics*, 99(6):323–329, 2012.

[126] K. Shaw. Biological applications of support vector machines.*Nature Education*, 1(1):201, 2008.

[127] Ribosomes, Transcription, and Translation. http://www.nature.com/scitable/topicpage/ribosomes-transcription-and-translation-14120660.

[128] C. Xing, D. L. Bitzer, W. E. Alexander, M. A. Vouk, and A. M. Stomp. Identification of protein-coding sequences using the hybridization of 18S rRNA and mRNA during translation.*Nucleic Acids Res.*, 37(2):591–601, Feb 2009.

[129] N. T. Ingolia, L. F. Lareau, and J. S. Weissman. Ribosome profiling of mouse embryonic stem cells reveals the complexity and dynamics of mammalian proteomes.*Cell*, 147(4):789–802, Nov 2011.

[130] M. Guttman, P. Russell, N. T. Ingolia, J. S. Weissman, and E. S. Lander. Ribosome profiling provides evidence that large noncoding RNAs do not encode proteins.*Cell*, 154(1):240–251, Jul 2013.

[131] Y. Arava, Y. Wang, J. D. Storey, C. L. Liu, P. O. Brown, and D. Herschlag. Genome-wide analysis of mRNA translation profiles in Saccharomyces cerevisiae.*Proc. Natl. Acad. Sci. U.S.A.*, 100(7):3889–3894, Apr 2003.

[132] Kozak M. Context effects and inefficient initiation at non-aug codons in eucaryotic cell-free translation systems.*Genome Research*, 9:5073–5080, 1989.

[133] Heng Xu and et. at al. Length of the ORF, position of the first AUG and the Kozak motif are important factors in potential dual-coding transcripts.*Cell Research*, 20:445–457, 2010.

[134] M. De Angioletti, G. Lacerra, V. Sabato, and C. Carestia. Beta+45 G –> C: a novel silent beta-thalassaemia mutation, the first in the Kozak sequence.*Br. J. Haematol.*, 124(2):224–231, Jan 2004.

[135] Kozak M. Initiation of translation in prokaryotes and eukaryotes.*Gene*, 234:187–208, 1999.

[136] Kozak M. Recognition of aug and alternative initiator codons is augmented by g in position +4 but is not generally affected by the nucleotides in positions +5 and +6.*EMBO J*, 16:2482–2492, 1997.

[137] Jörg Hackermüller2 Stephan H. Bernhart1 Peter F. Stadler1 3 4 Ulrike Mückstein1, Hakim Tafer1 and Ivo L. Hofacker1.*Bioinformatics*, 22(10):1177–1182, 2006.

[138] J. T. Vanselow A. Schlosser J. Vasquez, C. Hon and T. N. Siegel. Comparative ribosome profiling reveals extensive translational complexity in different Trypanosoma brucei life cycle stages.*Nucl. Acids Res.*, 42(6):3623–3637, 2014.

[139] R. D. Finn, J. Clements, and S. R. Eddy. HMMER web server: interactive sequence similarity searching.*Nucleic Acids Res.*, 39(Web Server issue):29–37, Jul 2011.

[140] M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E. L. Sonnhammer, S. R. Eddy, A. Bateman, and R. D. Finn. The Pfam protein families database.*Nucleic Acids Res.*, 40(Database issue):290–301, Jan 2012.

[141] Y. Zhang, Y. Sun, and J. R. Cole. A Sensitive and Accurate protein domain cLassification Tool (SALT) for short reads.*Bioinformatics*, 29(17):2103–2111, Sep 2013.

[142] M. Magrane and U. Consortium. UniProt Knowledgebase: a hub of integrated protein data. *Database (Oxford)*, 2011:bar009, 2011.

[143] S. R. Eddy. A new generation of homology search tools based on probabilistic inference. *Genome Inform*, 23(1):205–211, Oct 2009.

[144] Yuan Zhang and Yanni Sun. MetaDomain: a profile hmm-based protein domain classification tool for short sequences. In*Proceedings of Pacific Symposium on Biocomputing (PSB)*, 2012.

[145] Andy Liaw Chao Chen and Leo Breiman. Using Random Forest to Learn Imbalanced Data.

[146] Michael R. Lyu Kai-Zhu Huang, Haiqin Yang.*Machine Learning: Modeling Data Locally and Globally*. Springer Science and Business Media, 2008.

[147] Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka.

[148] Kim D. Pruitt, Tatiana A. Tatusova, and Donna R. Maglott. Ncbi reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins.*Nucleic Acids Research*, 35(Database-Issue):61–65, 2007.

[149] M. N. Cabili, C. Trapnell, L. Goff, M. Koziol, B. Tazon-Vega, A. Regev, and J. L. Rinn.

[150] Baris E. Suzek, Hongzhan Huang, Peter B. McGarvey, Raja Mazumder, and Cathy H. Wu. Uniref: comprehensive and non-redundant uniprot reference clusters.*Bioinformatics*, (10):1282–1288.

[151] PhyloCSF. https://github.com/mlin/PhyloCSF/wiki.

[152] T. R. Mercer, M. E. Dinger, and J. S. Mattick. Long non-coding RNAs: insights into functions.*Nat. Rev. Genet.*, 10(3):155–159, Mar 2009.