Grammar string: a novel ncRNA secondary structure representation

Rujira Achawanantakun, Seyedeh Shohreh Takyar, and Yanni Sun*

Department of Computer Science and Engineering, Michigan State University,
East Lansing, MI 48824 , USA
**Email: yannisun@cse.msu.edu*

Multiple ncRNA alignment has important applications in homologous ncRNA consensus structure derivation, novel ncRNA identification, and known ncRNA classification. As many ncRNAs' functions are determined by both their sequences and secondary structures, accurate ncRNA alignment algorithms must maximize both sequence and structural similarity simultaneously, incurring high computational cost. Faster secondary structure modeling and alignment methods using trees, graphs, probability matrices have thus been developed. Despite promising results from existing ncRNA alignment tools, there is a need for more efficient and accurate ncRNA secondary structure modeling and alignment methods.

In this work, we introduce **grammar 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). Being a string defined on a special alphabet constructed from a CFG, it converts ncRNA alignment into sequence alignment with $O(n^2)$ complexity. We align hundreds of ncRNA families from BraliBase 2.1 using grammar strings and compare their consensus structure with Murlet using the structures extracted from Rfam as reference. Our experiments have shown that grammar string based multiple sequence alignment competes favorably in consensus structure quality with Murlet. Source codes and experimental data are available at `http://www.cse.msu.edu/~yannisun/grammar-string`.

1. INTRODUCTION

Annotating noncoding RNAs (ncRNAs), which are not translated into protein but function directly as RNA, is highly important to modern biology. NcRNAs play diverse and important roles in many biochemical processes. For example, 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 [20]. Short interfering RNAs (siRNAs) involve in gene silencing in RNAi process [25].

Comparative ncRNA identification, which searches for ncRNAs through evidence of evolutionary conservation, is the state-of-the-art methodology for ncRNA finding. The functions of many types of ncRNA are determined not only by their sequences but also by their secondary structures, which describe base pair interactions in ncRNA sequences. For example, the cloverleaf structure is a prominent feature of tRNAs. Thus, comparative ncRNA identification must exploit both sequence and structural conservations. Stochastic context-free grammar (SCFG) [8] 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 [17, 39, 36], tree profiles [14, 13], 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 struc-

*Corresponding author.

ture 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-Altual statistics [21] 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 [41, 2, 24] 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 [14, 13, 17, 39, 36, 37] align predicted structured output by RNA folding tools. However, optimal prediction may not be the native structure [5], creating a need for choosing plausible structures as input to multiple alignment. In this work, we propose an efficient pattern matching method to preselect 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.

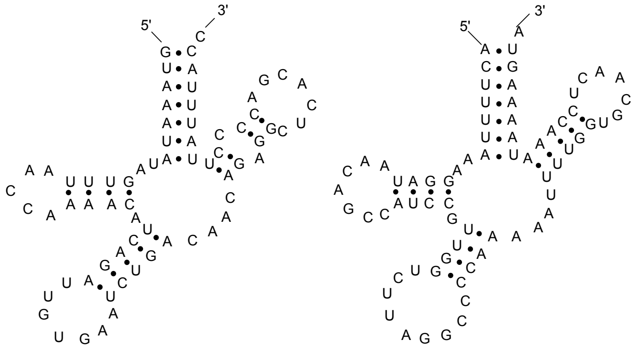
The remainder of the paper is organized as follows. Section 2 briefly introduces several representative secondary structure modeling methods, which will be compared to grammar strings in several experiments. Section 3 formally defines grammar string and illustrates its generation. Sections 4 and 5 present the algorithm and experiments of using grammar strings for multiple ncRNA alignment. Finally, we conclude this paper and discuss future directions in Section 6.

2. RELATED WORK

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 [32]. 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 [18], Consan [7], MARNA [34]. The second type of methods first builds a sequence alignment and then folds the alignment [16, 31, 38, 38, 23]. They infer structures from pre-aligned sequences generated using MULTIZ [3], ClustalW [35], 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 [30, 15]. NcRNA alignment is then converted into base pair probability matrix alignment. However, base pair probability matrix comparison is highly resource demanding. For example, pmcomp [17] 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 [39] and FOLDALIGNM [36] applied various restrictions or pruning techniques to reduce the time complexity. But they are still much more expensive than sequence alignment.



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

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

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

Fig. 1. Two tRNA sequences from the human genome and the alignment of their grammar strings.The stars below the alignment denote exact matches.

RNAforester [14, 13] 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 [39] 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.

3. APPROACH: GRAMMAR STRING DESIGN

Inspired by Jaakkola and Haussler's discriminative classification method [19], 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.

3.1. An unambiguous CFG for ncRNA generation

NcRNA structures without pseudo-knots can be derived by CFGs [8]. 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 $V \to \alpha$. 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 [6], using leftmost derivation. Following the general definition of a CFG, G4 has a finite set of nonterminal symbols $V = \{\mathcal{S}, \mathcal{T}\}$, a finite set of terminal symbols $T = \{A, C, G, U, \varepsilon\}$, and a finite set of production rules defined as below:

- $\mathcal{S} \to a\mathcal{S}|\mathcal{T}|\varepsilon$
- $\mathcal{T} \to \mathcal{T}a|a\mathcal{S}\hat{a}|\mathcal{T}a\mathcal{S}\hat{a}$

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 1, the following production rules from G4 are called: $\mathcal{S} \to \mathcal{T}$, $\mathcal{T} \to \mathcal{T}$ C, $\mathcal{T} \to$ G $\mathcal{S}$ C, $\mathcal{S} \to \mathcal{T}$, $\mathcal{T} \to$ U $\mathcal{S}$ A. Continuing to replace $\mathcal{S}$ 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's secondary structure in Figure 1. For a more detailed introduction about unambiguous CFGs, we refer readers to the review by Dowell and Eddy [6], where several light-weight unambiguous CFGs including G4 are discussed.

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

Thus, the new alphabet is $\mathcal{A} = \{$ 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 1 is: $\mathcal{S} \rightarrow \mathcal{T}$, $\mathcal{T} \rightarrow \mathcal{T}$ C, $\mathcal{T} \rightarrow$ G $\mathcal{S}$ C, $\mathcal{S} \rightarrow \mathcal{T}$, $\mathcal{T} \rightarrow$ U $\mathcal{S}$ A. Thus, the corresponding grammar string is "cPP" using the alphabet $\mathcal{A}$. 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 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 [1] 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);
}
```

Fig. 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 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, L).

3.3. 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 **grammar 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 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 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 [33] 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. We demonstrate the utility of grammar patterns in Section 4.3.

4. 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.

4.1. 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 [33] 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 $\mathcal{A}$. For each pair of characters a, b in $\mathcal{A}$, 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 ncRNA family database Rfam [12] 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[a]. 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.

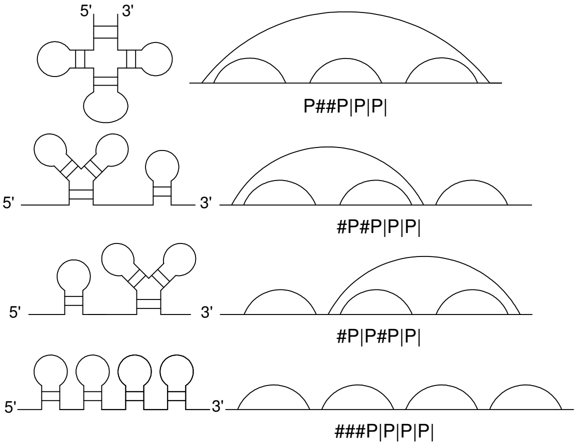


Fig. 3. Four different stem structures and their grammar patterns. The left column shows the 2D representation of an ncRNA folding. The right column shows the distributions of stems along an ncRNA sequence. All grammar patterns are generated using G4 (our chosen unambiguous context-free grammar).

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.

4.2. 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

[a] http://www.cse.msu.edu/~yannisun/grammar-string

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.

4.2.1. *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 [42], RNAstructure [29, 28], McCaskill's base pairing probability computation [30], etc. MFE-based methods can also be combined with other probabilistic models such as conditional log-linear models (CLLMs) in ContraFold [4] for structure prediction. In our experiments, we choose MFE based tool UNAFold [26, 27] 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 [5]. 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.

4.2.2. *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 [9]: $D = -\ln \frac{S_{real}(ij)-S_{rand}(ij)}{S_{iden}(ij)-S_{rand}(ij)}$, 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.

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

In our alignment pipeline, we allow multiple structures predicted for each input sequence, resulting in multiple grammar strings for a single ncRNA sequence. However, 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. In this section, we introduce an algorithm that uses grammar patterns introduced in Section 3.3 to pre-select grammar strings for multiple alignment.

UNAFold [26] 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 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.

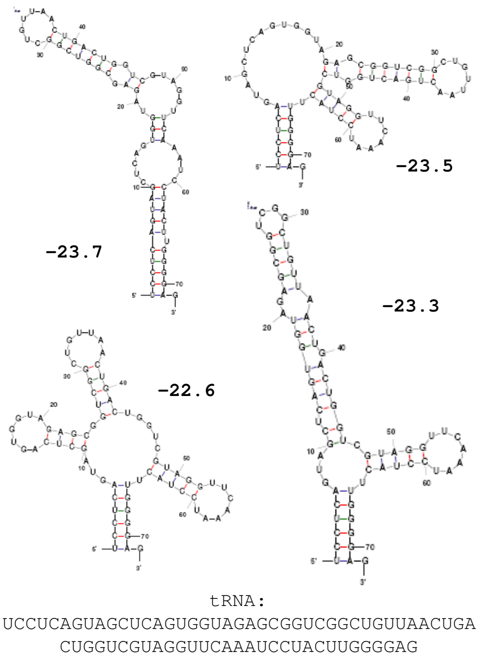

-23.7
-23.5
-23.3
-22.6
tRNA:
UCCUCAGUAGCUCAGUGGUAGAGCGGUCGGCUGUUAACUGA
CUGGUCGUAGGUUCAAAUCCUACUUGGGGAG

Fig. 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.

We conduct the screening step by choosing a representative structure favored by a majority of input sequences. In this step, we only examine the number of stems and their relative positions to each other. Insertions, deletions, or substitutions of bases or base pairs will be handled by alignment program. For example, if the cloverleaf structure is chosen as the preferred structure, only grammar strings encoding the same stem structure will be kept. Following the definition of grammar pattern in Section 3.3, we first simplify each grammar string as a grammar pattern, which encodes the stem structure of each ncRNA sequence. Then, we choose a grammar pattern that is shared by most input sequences and has the minimum sum of ΔGs. The assumption is that with multiple homologous ncRNA sequences available, the most popular structure will likely to be the native structure. Below, we elaborate the choice of the most popular structure using a voting mechanism.

(1) Convert each grammar string into a grammar pattern. Assign the computed thermodynamic energy ΔG of a grammar string to its grammar pattern. Suppose there are m input ncRNA sequences. N_i predictions are output for ncRNA sequence with index i. The output of this step is a set of grammar patterns and their associated ΔGs: $\{(s_1^1, \Delta G_1^1), (s_1^2, \Delta G_1^2), ..., (s_1^{N_1}, \Delta G_1^{N_1}), ..., (s_m^{N_m}, \Delta G_m^{N_m})\ \}$. s_i^j is the grammar pattern derived from the jth structure prediction for the ith input sequence. ΔG_i^j is the associated thermodynamic energy value for s_i^j.

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

$$f(s) = \sum_{i=1..m} \min_{j=1..N_i} \{\Delta G_i^j | s_i^j == s\} \qquad (1)$$

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

(3) Of multiple grammar strings generated for each ncRNA sequence, only keep the grammar string that can be converted to s^*. If more than one such grammar strings exists, keep the one with the minimum ΔG.

As tRNAs pose a hard case for MFE-based structure prediction programs [11, 12], we use a set of tRNA sequences as an example to illustrate how to choose the most favored stem structure for tRNAs. We randomly choose 20 tRNA sequences with average pairwise identity between 50% and 70% from BRAliBase III ncRNA sequence benchmark data set [10]. UNAFold is applied to each sequence allowing 5% sub-optimal structures predicted. More than one structure is predicted for each input sequence. After applying step 1, we summarize the grammar patterns, their encoded stem structures, and the corresponding ΔG for the first three tRNA sequences in Table 1. Note that we use a pair of symmetric brackets "()" to represent a stem. For example, "(()()())" encodes a cloverleaf structure with four stems. There are only three different grammar patterns in Table 1: $P\#\#P|P|P|$, $P\#P|P|$, and P. Following the definition of $f(s)$ in Equation 2, f($P\#\#P|P|P|$) is the sum of ΔGs of grammar patterns denoted with *. Therefore, f($P\#\#P|P|P|$) = -71.3. f(P) and f($P\#P|P|$) are much larger than f($P\#\#P|P|P|$). Thus, $P\#\#P|P|P|$ is the consensus stem structure for the three tRNA sequences. Note that no grammar string is chosen for "seq 3" because none of them is equal to the consensus stem structure.

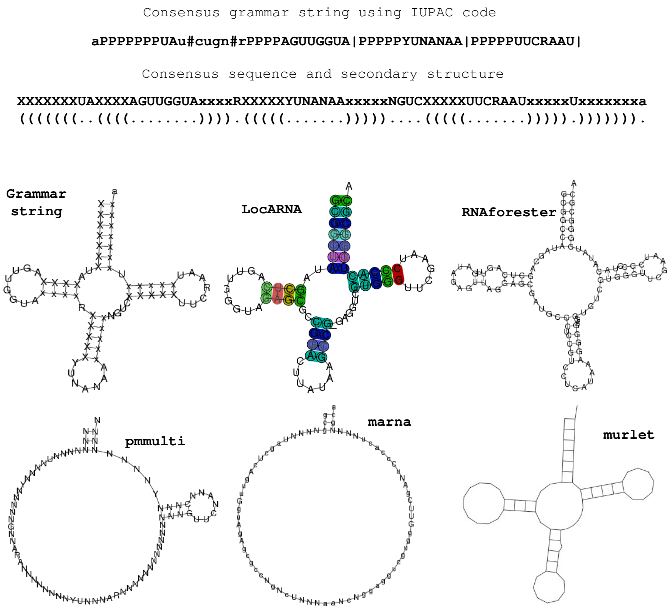
Table 1. Structure predictions for three tRNA sequences. Multiple structure predictions are output for each sequence. For each prediction, column named "stems" displays its stem structure denoted by brackets. The corresponding grammar pattern and ΔG are listed in columns 3 and 4, respectively.

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

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.

5. EXPERIMENTAL 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 4.3. Figure 5 shows the consensus secondary structure derived from aligning grammar strings of given tRNAs. We also tested other structural alignment programs including pmmulti [17], Murlet [22], RNAforester [14, 13], MARNA [34], and LocARNA [39]. Figure 5 shows that the grammar string alignment and Murlet both generate the best consensus structure for tRNA sequences.



Consensus grammar string using IUPAC code

```
aPPPPPPPUAu#cugn#rPPPPAGUUGGUA|PPPPPYUNANAA|PPPPPUUCRAAU|
```

Consensus sequence and secondary structure

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

Fig. 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.

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 [40]. 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 [26, 27] on over 6700 sequences from these 452 families. The average number of 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. And, the average time to fold 15 sequences in each family is 7 seconds on a Core 2 Duo 1.7GHz laptop. The average time to align 15 grammar strings is 3 seconds.

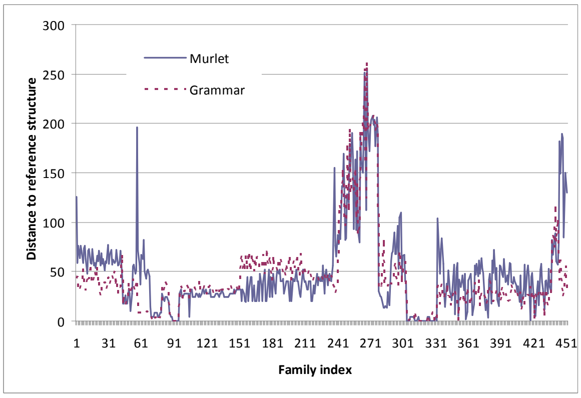


Fig. 6. The differences of the reference structures (from Rfam) and the predicted consensus structures from grammar string and Murlet alignments are plotted and compared. Lower numbers indicate higher similarity between the predicted structure and the reference structure.

As Murlet [22], 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 5). And then this consensus grammar string is translated into a secondary structure using a reversed protocol to the one described in Figure 2. Murlet outputs the consensus structure along with each alignment. We compare the predicted secondary 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 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.

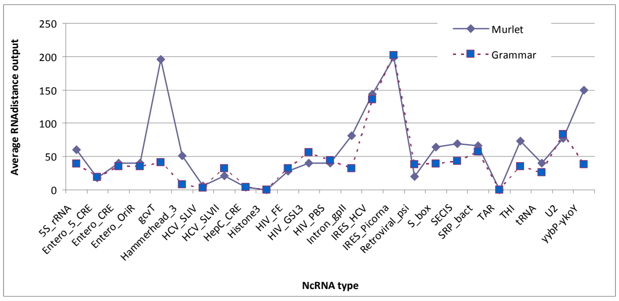


Fig. 7. Consensus structures are derived for multiple families of each type of ncRNA, resulting a RNAdistance output vector. For each type of ncRNA, the average RNAdistance output for Murlet and grammar string alignment is compared.

In order to analyze how grammar string and Murlet perform on each type of ncRNA, Figure 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 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.

6. CONCLUSION AND FUTURE WORK

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.

References

1. 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.
2. 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.
3. 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.
4. Chuong B. Do, Daniel A. Woods, and Serafim Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–98, 2006.
5. Kishore Doshi, Jamie Cannone, Christian Cobaugh, and Robin Gutell. Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. *BMC Bioinformatics*, 5(1):105, 2004.
6. Robin Dowell and Sean Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5(1):71, 2004.
7. Robin D Dowell and Sean R Eddy. Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinformatics*, 7(400), 2006.
8. 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.
9. 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.
10. E.K. Freyhult, J.P. Bollback, and P.P. Gardner. Exploring genomic dark matter: a critical assessment of the performance of homology search methods on noncoding RNA. *Genome Research*, 17(1):117–125, 2006.
11. S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy. Rfam: an RNA family database. *Nucleic Acids Res.*, 31(1):439–441, 2003.
12. 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.
13. 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.
14. 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.
15. I.L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Res.*, 31:3429–3431, 2003.
16. IL Hofacker, M Fekete, and PF Stadler. Secondary

structure prediction for aligned RNA sequences. *J Mol Biol.*, 319(5):1059–66, 2002.

17. Ivo L. Hofacker, Stephan H. F. Bernhart, and Peter F. Stadler. Alignment of RNA base pairing probability matrices. *Bioinformatics*, 20(14):2222–2227, 2004.
18. Ian Holmes. Accelerated probabilistic inference of rna structure evolution. *BMC Bioinformatics*, 6(1):73, 2005.
19. Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, pages 487–493, 1998.
20. 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.
21. 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):22642268, 1990.
22. 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.
23. B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res*, 31(13):3423–3428, 2003.
24. C Li, A H Wang, and L Xing. Similarity of RNA secondary structures. *J. Comput. Chem.*, 28:508–512, 2007.
25. Shanfa Lu, Rui Shi, Cheng-Chung Tsao, Xiaoping Yi, Laigeng Li, and Vincent L. Chiang. RNA silencing in plants by the expression of siRNA duplexes. *Nucl. Acids Res.*, 32(21):e171–, 2004.
26. N. R. Markham and M. Zuker. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Res.*, 33:W577–W581, 2005.
27. N. R. Markham and M. Zuker. UNAFold: software for nucleic acid folding and hybriziation. In *Bioinformatics, Volume II. Structure, Functions and Applications, number 453 in Methods in Molecular Biology*, pages 3–31, Totowa, NJ, USA, 2008. Humana Press.
28. 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.
29. 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 algorothm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences USA*, 101:7287–7292, 2004.
30. J. S. Mccaskill. The equilibrium partition function and base pair binding probabilities for rna secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990.
31. Elena Rivas and Sean R. Eddy. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, 2(1):8, 2001.
32. David Sankoff. Simultaneous Solution of the RNA Folding, Alignment and Protosequence Problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825, 1985.
33. 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.
34. 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.
35. 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.
36. Elfar Torarinsson, Jakob H. Havgaard, and Jan Gorodkin. Multiple structural alignment and clustering of RNA sequences. *Bioinformatics*, 23(8):926–932, 2007.
37. Helene Touzet and Olivier Perriquet. CARNAC: folding families of related RNAs. *Nucl. Acids Res.*, 32(suppl. 2):W142–145, 2004.
38. 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.
39. 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.
40. A. Wilm, I Mainz, and G. Steger. An enhanced rna alignment benchmark for sequence alignment programs. *Algorithms Mol Biol.*, 1(19), 2006.
41. Y Zhang, J Qiu, and Su L. Comparing RNA secondary structures based on 2D graphical representation. *Chemical Physics Letters*, 458:180–185, 2008.
42. 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.