

Consensus Folding of Unaligned RNA Sequences Revisited

Vineet Bafna¹, Haixu Tang², and Shaojie Zhang¹

¹ Dept. of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093. vbafna@cs.ucsd.edu, shzhang@cs.ucsd.edu

² School of Informatics and Center for Genomics and Bioinformatics, Indiana University, Bloomington, IN 47408. hatang@indiana.edu

Abstract. As one of the earliest problems in computational biology, RNA secondary structure prediction (sometimes referred to as “RNA folding”) problem has attracted attention again, thanks to the recent discoveries of many novel non-coding RNA molecules. The two common approaches to this problem are *de novo* prediction of RNA secondary structure based on energy minimization and “consensus folding” approach (computing the common secondary structure for a set of unaligned RNA sequences). Consensus folding algorithms work well when the correct seed alignment is part of the input to the problem. However, seed alignment itself is a challenging problem for diverged RNA families.

In this paper, we propose a novel framework to predict the common secondary structure for unaligned RNA sequences. By matching putative stacks in RNA sequences, we make use of both primary sequence information and thermodynamic stability for prediction at the same time. We show that our method can predict the correct common RNA secondary structures even when we are only given a limited number of unaligned RNA sequences, and it outperforms current algorithms in sensitivity and accuracy.

1 Introduction

With the recent discovery of novel non-coding RNA (*ncRNA*) families, RNA is rapidly gaining importance as a molecule of interest [1, 2]. Recent article puts the number of human genes down to about 20000 – 25000 [3]. By comparison, even the worm *C. elegans* has around 19,500 genes. On the other hand, expression activity has been detected on a much larger portion of human genome [4]. It is likely that many of these novel transcripts are ncRNA genes, which may carry on many unknown cellular functions. Like proteins, RNA structures are more important for function than their sequences: RNA with similar function often have similar structures but distinct primary sequences. Therefore, understanding the structures of these RNA molecules will help elucidate their functions. Consider the recent exciting discovery of riboswitches [5, 6] as an example. These control elements, with a conserved secondary structure, are located in the untranslated

regions of genes coding for proteins that are involved in variant metabolite (nucleic acids, amino acids etc.) synthesis pathways. The riboswitches can turn off the expression of their downstream genes by binding to certain metabolites, subsequently changing their secondary structures and blocking the translation initiation.

While there is a resurgence in interest in ncRNA, the problem of RNA secondary structure prediction has been extensively studied since the 70s. The key idea is that to stabilize its structure, distant base-pairs in the single stranded RNA molecule must form hydrogen bonds. There are two distinct approaches to predict RNA secondary structure. The RNA folding approach, initiated by Tinoco et al. [7], assigns free energies to the components of RNA secondary structure, and then computes the RNA secondary structure with the minimum energy. Dynamic programming algorithms have been developed to compute minimum energy secondary structures [8–12], and implemented in software packages such as MFOLD [13] and ViennaRNA [14]. However, RNA folding via energy minimization has its shortcomings. First, fold prediction depends critically upon correct values of the energy parameters, as shown by Jaeger et al. [15], which are hard to obtain experimentally. Also, RNA folding in a real cell is mediated by interactions with other molecules, and the absence of knowledge of these interactions may cause mis-folding *in silico*. Pavesi et al. [16] tried to alleviate this problem by comparing minimum energy structures of a set of RNA sequences from the same family to determine conserved secondary structure. However, it is unclear how the misprediction of secondary structure for a single RNA sequence can affect the accuracy of this approach.

A different approach attempts to resolve these shortcomings by using evolutionary conservation of structure as the basis for structure prediction. It needs as input multiple RNA sequences from an RNA family, which have common secondary structures. Since for divergent sequences, the mutations in base-pairing regions must be compensated in the complementary base to preserve structure, the presence of multiple covarying mutations is a strong signal for base-pairing. In fact most RNA sequences are selected more for maintenance of the structure than conservation of primary sequence. If the sequence similarity between the given RNA sequences is appropriate, one can first align these sequences using multiple sequence alignment algorithm and then figure out the potential base pairs in RNA secondary structures by looking at regions with a high number of compensating mutations. Levitt successfully derived the theoretical tRNA secondary structure using this approach, which was largely confirmed by crystallography [17], and various other structures have been determined through such analysis. Computer programs were implemented later on to achieve this goal automatically [18].

However, aligning multiple and divergent RNA sequences so as to preserve their conserved structures is not easy, because many compensatory mutations decrease the overall sequence similarity. For unaligned sequences, one must compute the structure and alignment simultaneously. Sankoff proposed an algorithm that can simultaneously align RNA sequences and find the optimal common

fold [19–21]. However, the complexity of this algorithm is $O(l^6)$, where l is the length of RNA sequences, too high to be practical even for two sequences. The complexity can be reduced to $O(l^4)$ [22], but only when RNA has no multi-loop structure. Eddy and Durbin, and other groups [23–25] pioneered the approach of modeling RNA sequences using stochastic context free grammars. The rules of the SCFG allow for position dependent scoring of distant base-pairs and primary sequence conservation, and also allow automated estimation of model parameters from unaligned sequences using EM. However, in practice, the extensive divergence of RNA sequences makes it hard to reconstruct structure and alignment with perfect accuracy, and the covariance models work best when supplied with good seed alignments. Much recent work has focused on improving fold prediction for aligned sequences [18, 26].

In our approach to this well-researched problem, we are motivated by the idea of constraining allowed folds to make it more likely to reach the final correct structure. This idea has been used with good success in aligning divergent genomic sequences. For diverged DNA sequences, there is not enough signals for probabilistic models such as HMMs to be effective without prior information (in the form of seed alignments). The recent methods [27] identify anchors corresponding to highly conserved orthologous regions, and use these to constrain the multiple alignments. This approach has been used in RNA as well. Waterman [28] pioneered this with a statistical approach for choosing conserved stem-loops within a pair of fixed size windows in a set of unaligned RNA sequences. Ji and Stormo [29] extended this idea considerably. Starting with putative stem-loops, they remove all but the ones conserved in a global sequence alignment of two sequences. These are further culled to retain only those that are present in every sequence in the family. Perriquet et al. [30] proposed a different anchoring approach to solving the same problem by first determining anchor regions that are highly conserved in given RNA sequences and then seeking a set of conserved stems crossing the same anchor regions that have minimum folding energy. Both methods use primary sequence conservation extensively, and limit the variability in the length of loop regions, which may lead to accurate but relatively few predicted anchor stacks for diverged families. On the contrary, Bouthinon and Soldano [31], Davydov and Batzoglou [32] each proposed algorithms to select conserved base-pairs among the given RNA sequences based solely on the conservation of the structure that they form, considering neither their sequence similarity nor the thermodynamic stability of this structure. As a result, these methods may risk selecting wrong base-pairs when only a limited number of RNA sequences are given.

In this paper, we describe a method, **RNAScf** (RNA Stack based Consensus Folding), for predicting the consensus fold of an RNA family, given unaligned sequences. Our method is based on the notion of finding structurally conserved anchors, and an iterative extension constrained by the anchors. With relatively few parameters, and limited training, the method outperforms other competing methods (See RESULTS), detecting 88% of true stacks (sensitivity), and overlapping with a correct stack in 93% of all predictions. The method establishes

the validity of this new paradigm for computing consensus structure in RNA. We also discuss extensions based on an iterative refinement of the predicted structure. (See DISCUSSION).

2 Approach

As shown in Figure 1(a), the secondary structure of an RNA has a tree like shape. Assume that there is a dummy base-pair between 0 and $n + 1$. Define a *loop* as a set of indices $i_1 \leq i_2 \dots \leq i_k$ such that for all j , either $i_{j+1} = i_j + 1$, or (i_j, i_{j+1}) form a base-pair. Further, if for some j, j' , $(i_j, i_{j'})$ form a base-pair, then $|j - j'| = 1$. It can be seen that the structure can be decomposed uniquely into a set of loops, and the loops can be classified as *hairpin* (containing only one base-pair), a *stem-loop* (two base-pairs with no unpaired bases), an *interior loop/bulge* (two base-pairs with unpaired bases), and a *multi-loop* ($k > 2$ base-pairs).

The stability of the RNA structure is determined predominantly by *stacks* of consecutive stem-loops. The stacks are stabilized by hydrogen bonds between the base-pairs, and in general, the longer a stack region, the more energetically favorable it is. Each stack corresponds to a pair of sub-strings. These pairs are typically non-interleaving. While interleaved stacks, or *pseudo-knots* (such as the pair (f, f') , and (h, h') in Figure 1(a)) do occur, they are relatively less common and ignored here. In our approach for finding anchors, we ignore individual base-pairs and work with a slightly generalized notion of a stack that includes unpaired bases. Configurations of stacks that are conserved in multiple sequences will be the anchors in determining consensus structures.

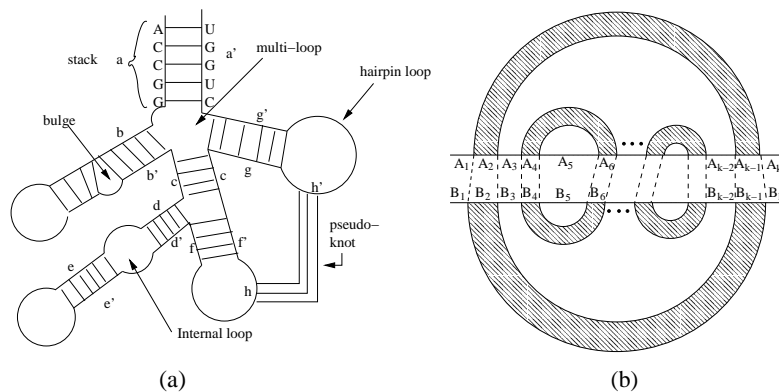


Fig. 1. (a) An RNA secondary structure structure with various structural elements including stacked stem-loops, bulges, hairpins, and multi-loops. (b) Two stack configurations match to each other for both unpaired regions and paired regions.

2.1 Predicting Putative Stacks

The thermodynamic stability of a stack is proportional to the number of hydrogen bonds between the base-pairs in the stack. Any pair of strings can be aligned (with gaps) so as to optimize the energy of the paired bases. Therefore, given an RNA string A , we construct a local alignment of $A[1, \dots, n]$ with $A[n, \dots, 1]$. Let $\delta_h(i, j)$ be the score (number of h-bonds) in an $(A[i], A[j])$ base-pairing. Thus base-pairing of **G-C**, **A-T**, **G-U**, is scored 3, 2, and 1 respectively. Let $S[i, j]$ be optimum score for a stack with left end-point i , and right end-point j . Then

$$S[i, j] = \max \begin{cases} S[i+1, j-1] + \delta_h(i, j) & (\delta_h[i, j] > 0) \\ S[i+1, j] + g \\ S[i, j-1] + g \\ 0 \end{cases} \quad (1)$$

where g is a gap penalty. In our implementation, we modify this basic approach to include affine gap costs. We select each (i, j) for which $S[i, j]$ is greater than some threshold. In order to avoid predicting overlapping stacks, we sort the stacks by decreasing score values. Each time a stack is picked, all base-pairs in it are excluded. While straightforward, this is an effective procedure. Intuitively the probability of finding a base-pair at random is much higher than the probability of finding a high-scoring stack. Waterman showed that finding a k -long stack within certain window size in many given sequences (even not all) can be significant [28]. Also, most real base-pairs in correct structures should be stabilized by multiple stacked base-pairs, implying that limiting consideration to high-scoring stacks does not result in many false negatives. To test this, we computed statistics on the seed-alignments in the RFAM database [33]. All stacks from seed alignments of 379 families (9559 sequences) were selected. The seed alignment contains known representative members of the family, which is hand-curated and is annotated with structural information. To correct for annotation errors, the stacks were realigned to each other locally, and extended so as to be locally maximal without unpaired bases. Figure 2 describes plots the cumulative distributions of stacks according to the minimum number of hydrogen bonds (Figure 2(a)), or stack length (Figure 2 (b)). Additionally, we also plotted the number of putative stacks that can be found on RFAM sequences (normalized to a 100bp region). If all possible base-pairs were considered, we see about 900 putative stacks in a 100bp region. This number grows quadratically with the length of the sequence increases. Instead, by limiting attention to stacks with length greater than 4, *and* at least 8 hydrogen bonds, we have only 34 putative stacks in a 100bp region. At the same time, we miss only a small fraction of the true stacks. This computation shows that in considering anchors, it is reasonable to restrict attention to longer stacks.

2.2 Stack Configurations

A putative stack of length 4 can still be found by chance. Ji and Stormo [29] and Perriquet et. al [34] both use sequence similarity to further select stacks.

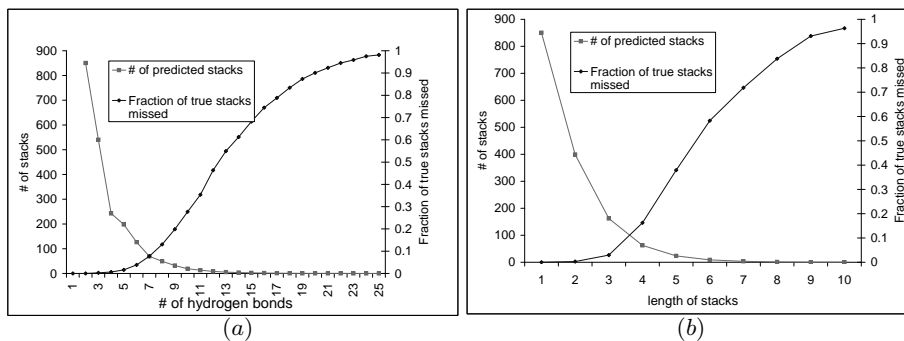


Fig. 2. Statistics of the stacks in Rfam database. (a) One line shows the fraction of annotated stacks which would be missed out by using different cutoff number of hydrogen bonds. The other line shows the number of predicted stacks per 100 based all Rfam seed sequences for each cutoff number of hydrogen bonds. (b) One line shows the fraction of annotated stacks which would be missed out by using different cutoff length of the stacks. The other line shows the number of predicted stacks per 100 bases for each cutoff length of stacks. We also can combine two cutoffs together - the length of the stacks and the number of hydrogen bonds- to achieve better performance.

We propose to select a set of stacks instead of one at a time. We evaluate the set of stacks by both the stability (free energy) of the structure they form and the sequence similarity computed based on these common stacks as anchors. To describe evaluation function of the selected set, we must define a notion of *configuration* of stacks. Note that two stacks P_1 and P_2 may have one of the following relations [31]: (1) P_1 and P_2 are interleaving; (2) P_1 is enclosed within P_2 (denoted by $P_1 <_I P_2$); (3) P_1 is juxtaposed to P_2 so that its right end-point is before the left end-point of P_2 ($P_1 <_p P_2$). An RNA structure R_A on A is a collection of non-interleaving stacks on A . The stacks have a partial order ($<_I \cup <_p$) defined on them. The free energy of the structure is the sum of free energies of each of the constituent hairpins, stem-loops, interior loops, and multi-loops. Here, we work with the notion of generalized stacks instead of stem-loops. The energy of a stack P denoted by $\mathcal{E}_s(P)$ is as described in equation 1. The energies of loops \mathcal{E}_h (hairpin), \mathcal{E}_b (bulge), \mathcal{E}_m (multi-loops), are a function of the length of the sequence and set as described in Jaeger et al. [15].

Define a *consensus* structure $\mathcal{P}(A, B)$ as a pair of structures R_A and R_B on A and B respectively, with a one-one correspondence between stacks in R_A and R_B such that the corresponding stacks in the two structures maintain identical partial order relationships. We define the free energy $\phi(\mathcal{P}(A, B))$ of the consensus structure similar to the energy of the individual structures. For each pair of corresponding stacks, or loops, the maximum of the two energy values is chosen.

Given a consensus $\mathcal{P}(A, B)$, the sequences A and B can be aligned to be consistent with $\mathcal{P}(A, B)$ (see Figure 1(c)), so that the sequences in stacks are

aligned to each other, and likewise for the sequences in the unpaired regions. This alignment partitions sequence A and B into alternating stack and non-stack regions A_1, A_2, \dots, A_k , and B_1, B_2, \dots, B_k . Each pair of sequences (A_i, B_i) , is aligned optimally. We define such an alignment as a *configuration*. The cost of the configuration $(A, B, \mathcal{P}(A, B))$ is defined as a function of sequence similarity and the energy of the consensus structure. Denote $i \in \mathcal{P}(A, B)$ if and only if (A_i, B_i) are paired in a stack with some (A_j, B_j) in $\mathcal{P}(A, B)$. Let $\mathcal{S}(A_i, B_i)$ denote the cost of an optimal global alignment of subsequences A_i and B_i . The *cost* of the configuration $(A, B, \mathcal{P}(A, B))$ is denoted by

$$M(\mathcal{P}(A, B)) = w_1 \Phi(\mathcal{P}(A, B)) + w_2 \sum_{i \in \mathcal{P}(A, B)} \mathcal{S}(A_i, B_i) + w_3 \sum_{i \notin \mathcal{P}(A, B)} \mathcal{S}(A_i, B_i) \quad (2)$$

where $w_1 + w_2 + w_3 = 1$, represent parameters describing the relative weights to the free energy of the configuration, sequence similarity in stack regions, and sequence similarity within loop regions. Ideally, these weights should be adjusted according to the number and divergence of the given sequences. However, in the tests through this paper, we use an identical set of weights ($w_1 = 0.84$, $w_2 = 0.06$, and $w_3 = 0.1$). For a given pair, we compute consensus structures of minimum cost.

The definition of a configuration of stacks for a pair of sequences also extends to multiple sequences: A *configuration* $\mathcal{P}(A_1, A_2, \dots, A_s)$ is a collection of s RNA structures $\{P^{A_1}, P^{A_2}, \dots, P^{A_s}\}$, one for each sequence with the following property: For each pair of structures, there is a one-one correspondence between the stacks that are consistent with the partial orders $<_I$ and $<_P$. A configuration with l stacks partitions each sequence A_i into $2l + 1$ blocks denoted $A_{i,1}, A_{i,2}, \dots, A_{i,2l+1}$, where each block $A_{*,j}$ is either a stack in the configuration ($j \in \mathcal{P}$), or part of the loop region. We modify equation 2 to describe the cost of the configuration $\mathcal{P}(A_1, A_2, \dots, A_s)$, as follows:

$$M(\mathcal{P}(A_1, \dots, A_s)) = w_1 \Phi(\mathcal{P}(A_1, \dots, A_s)) + w_2 \sum_{j \in \mathcal{P}} \mathcal{S} \begin{pmatrix} A_{1,j}, \\ A_{2,j}, \\ \dots, \\ A_{s,j} \end{pmatrix} + w_3 \sum_{j \notin \mathcal{P}} \mathcal{S} \begin{pmatrix} A_{1,j}, \\ A_{2,j}, \\ \dots, \\ A_{s,j} \end{pmatrix} \quad (3)$$

Here the function \mathcal{S} computes the score of a multiple alignment. The *RNA stack based consensus folding problem* can be described formally: given s RNA sequences, compute a minimum cost *stack configuration*. In the following section, we describe algorithms for computing optimal configurations.

3 Stack based Consensus Folding

3.1 Computing optimal stack configuration in two RNA sequences

We use dynamic programming to compute an optimal configuration. The algorithm is similar to prior work [35, 20] with an important difference being that stacks (instead of individual base-pairs) are now used. Given sequences

A, B , we compute all potential stacks in them, using the algorithm from Section 2.1. Assume these two sequences have m and n stacks respectively. Let $\mathcal{P}^A = P_1^A, P_2^A, \dots, P_m^A$ and $\mathcal{P}^B = P_1^B, P_2^B, \dots, P_n^B$ denote the stacks, ordered according to increasing values of the right-most base-pair. Denote the index of the first and last base pair of a stack P as P_b, P_e , and the length as P_l . Define the following terms:

Seq(P^A): The sub-sequence covered by the stack P^A , given by $A[P_b^A \dots P_b^A + P_l^A - 1]$, and $A[P_e^A - P_l^A + 1 \dots P_e^A]$.

Loop(P^A): The sub-sequence covered by the first and last positions of the stack P^A after excluding the bases in $\text{Seq}(P^A)$. In other words, the sequence $A[P_b^A + P_l^A \dots P_e^A - P_l^A]$.

Loop($P^A, P^{A'}$): If $P^{A'}$ is enclosed within P^A , then the loop region corresponds to the sequence in between the two stacks (i.e., the subsequences $A[P_b^A + P_l^A \dots P_b^{A'} - 1]$, and $A[P_e^{A'} + 1 \dots P_e^A - P_l^A]$). If $P^{A'}$ is to the left of P^A , the loop region corresponds to $A[P_e^{A'} + 1 \dots P_b^A - 1]$. Otherwise, the term is undefined.

$M(P^A, P^B)$: The cost of an optimum configuration of A and B over all consensus structures, given that stacks P^A and P^B are in the consensus structure and aligned to each other.

Clearly, it is sufficient to compute $M[P^A, P^B]$ for all pairs in $\mathcal{P}^A \times \mathcal{P}^B$, which would need $O(m^2n^2)$ time. In computing $M[P^A, P^B]$, we have 3 choices for the subsequences $\text{Loop}(P^A)$, and $\text{Loop}(P^B)$, as they could either form a hairpin, an interior loop/bulge, or a multi-loop. Therefore,

$$M[P^A, P^B] = M_s[P^A, P^B] + \min \left\{ \begin{array}{l} M_h[P^A, P^B], \text{ (* hairpin loop *)} \\ M_b[P^A, P^B], \text{ (* interior loop/bulge *)} \\ M_m[P^A, P^B] \text{ (* multi-loop *)} \end{array} \right\} \quad (4)$$

here, $M_s(P^A, P^B)$ is the score matching stacks P^A and P^B , based on sequence and structure conservation, and can be computed by

$$M_s[P^A, P^B] = w_1 \max \left\{ \begin{array}{l} \mathcal{E}_s(P^A), \\ \mathcal{E}_s(P^B) \end{array} \right\} + w_2 \mathcal{S} \left(\begin{array}{l} \text{Seq}(P^A) \\ \text{Seq}(P^B) \end{array} \right) \quad (5)$$

$M_h[P^A, P^B]$ is the score of the loop regions of P^A and P^B given that no other matched stack pair is included by P^A and P^B i.e. these regions form matched hairpin loops.

$$M_h[P^A, P^B] = w_1 \max \left\{ \begin{array}{l} \mathcal{E}_h(|\text{Loop}(P^A)|), \\ \mathcal{E}_h(|\text{Loop}(P^B)|) \end{array} \right\} + w_3 \mathcal{S} \left(\begin{array}{l} \text{Loop}(P^A) \\ \text{Loop}(P^B) \end{array} \right) \quad (6)$$

$M_b[P^A, P^B]$ represents the matching score when P^A , and P^B are followed by an interior loop, or bulge. Consider all stacks P^x, P^y that are enclosed by P^A , and P^B , respectively. Then, $M_b(P^A, P^B)$ is the minimum free energy of any

matching of P^x, P^y .

$$M_b[P^A, P^B] = \min_{\substack{P^x <_I P^A, \\ P^y <_I P^B}} \left\{ w_1 \max \begin{cases} \mathcal{E}_b(|\text{Loop}(P^x, P^A)|), \\ \mathcal{E}_b(|\text{Loop}(P^y, P^B)|) \end{cases} \right. \\ \left. + w_3 \mathcal{S} \left(\begin{matrix} \text{Loop}(P^x, P^A), \\ \text{Loop}(P^y, P^B) \end{matrix} \right) + M[P^x, P^y] \right\} \quad (7)$$

For the multi-loop case, we need to define some additional terms. A sequence of stacks P_1, P_2, \dots form a *chain* if $P_1 <_p P_2 <_p \dots$. $M_m[P^A, P^B]$ represents the matching score between P^A , and P^B , given that there is a pair of chains included by P^A and P^B that form the multiloop. Let P_1^A, P_2^A, \dots (respectively P_1^B, P_2^B, \dots) denote stacks enclosed by P^A (P^B , respectively) and ordered according to increasing values of the last coordinate. Denote $P_{i_1}^A \in F(P_{i_2}^A)$ if $P_{i_1}^A <_p P_{i_2}^A$ and there is no stack P_j^A such that $P_{i_1}^A <_p P_j^A <_p P_{i_2}^A$. Then

$$M_m[P^A, P^B] = \min_{i,j} \{ M_c[P_i^A, P_j^A] + \max \begin{cases} \mathcal{E}_m(|\text{Loop}(P_i^A, P^A)|), \\ \mathcal{E}_m(|\text{Loop}(P_j^B, P^B)|) \end{cases} \} \quad (8)$$

Here, $M_c[P_i^A, P_j^B]$ is defined as the minimum energy of a chain that ends at P_i^A , and P_j^B , and begins at some $P_{i'}^A <_p P_i^A$, and $P_{j'}^B <_p P_j^B$. (The end conditions are added for efficiency reasons). Then,

$$M_c[P_i^A, P_j^B] = \min_{\substack{P_x^A \in F(P_i^A) \\ P_y^B \in F(P_j^B)}} \left\{ \begin{matrix} M_c[P_x^A, P_y^B] + M_o[P_x^A, P_i^A; P_y^B, P_j^B] + M[P_i^A, P_j^B], \\ M_c[P_x^A, P_y^B] + \mathcal{E}_m(|\text{Loop}(P_x^A, P_i^A)| + |\text{Seq}(P_i^A)|), \\ M_c[P_x^A, P_y^B] + \mathcal{E}_m(|\text{Loop}(P_y^B, P_j^B)| + |\text{Seq}(P_j^B)|) \end{matrix} \right\} \quad (9)$$

where $M_o[P_x^A, P_i^A; P_y^B, P_j^B]$ is the minimum free energy of the matching between the loops (P_x^A, P_i^A) and (P_y^B, P_j^B) ,

$$M_o[P_x^A, P_i^A; P_y^B, P_j^B] = w_1 \max \begin{cases} \mathcal{E}_b(|\text{Loop}(P_x^A, P_i^A)|), \\ \mathcal{E}_b(|\text{Loop}(P_y^B, P_j^B)|) \end{cases} + w_3 \mathcal{S} \left(\begin{matrix} \text{Loop}(P_x^A, P_i^A), \\ \text{Loop}(P_y^B, P_j^B) \end{matrix} \right) \quad (10)$$

3.2 Consensus fold computation for multiple RNA sequences

The optimal configuration of a randomly chosen pair of sequences from a family already shows high sensitivity (data not shown). It is likely that an optimal configuration of structures conserved in diverse multiple sequences will be very accurate. Recall the cost of the configuration as Equation (3), where \mathcal{S} denotes the score of a multiple alignment of the block. Clearly, the problem of computing optimal configuration is hard, given the discussion for the pairwise case. Here, we use a heuristic principle based on the notion of a star-alignment, with a seed configuration chosen from an optimal configuration of a random pair of sequences. To understand why our approach should work, we describe a back of the envelope calculation.

Consider a stack x of length k from the seed structure defined on sequence A_1 that is in fact incorrect (does not overlap with a true stack). For x to be

retained in the final anchored configuration, it must match with stacks in a large fraction of the other sequences. Let p be the probability that a random pair of bases can form base-pairs. Given an interval (i, j) in some sequence, the probability that (i, j) is the end of a stack of length k is p^k . Then, the probability that x is matched up with some random set of base-pairs defined by the end-points (i, j) is no more than p^k , even after ignoring sequence similarity. However, x cannot be matched to any other arbitrary stack. As we also score for primary sequence conservation, and the match should maintain the partial order of the configuration, (i, j) and x must be 'similarly situated'. To model this, we introduce a parameter w . Define w as the number of distinct pairs (i, j) such that x is allowed to match to. Then, the probability that x finds a match by chance is given by $p_x = 1 - (1 - p^k)^w$. Allowing a more flexible definition, we say that x is f -conserved in the configuration if it finds a match in at least $(1 - f)s$ of the s sequences.

$$Pr[x \text{ is } f\text{-conserved}] = P_c(x) = \sum_{l \geq (1-f)s} \binom{s}{l} p_x^l (1 - p_x)^{s-l} \quad (11)$$

Fix some parameters as follows. Let $p = \frac{3}{8}$ (corresponding to G-C, G-U, A-U), $f = 0.7$. The probability of getting an incorrect conserved stack depends critically on the parameters w . If w is too large, there is a high probability of getting random stacks to match up ($P_c(x) = 0.91$, when $k = 4, w = 80$ and $s = 20$). This effect might be offset by increasing k ($P_c(x) = 1.8e - 6$, when $k = 5, w = 80$ and $s = 20$), but then we risk losing many true (smaller size) stacks, which may cause incorrect pairs to be matched. The effect is also offset (to a less degree) by increasing the number of sequences, but that may not always be possible. The reason our approach works is because the choice of a conserved configuration restricts the possible stacks that x can match, effectively keeping the value of w low.

Before describing the approach, we must first modify the formulation to allow stacks to be *partially conserved*, and therefore, absent from some sequences. For $0 < f \leq 1$, define an f -configuration as a configuration with the following property: for every set of s corresponding stacks (one from each sequence), at most $(1 - f)s$ can be absent. In Figure 3, we describe a procedure for computing an anchor configuration.

The anchor configuration consists of stacks that optimize the cost of the configuration, and are conserved across the family. Thus, the stacks are highly likely to be correct. However, the procedure might also miss some true stacks due to a high initial value of k , and requirement of conservation. To increase sensitivity, we now search for less conserved, and shorter stacks. However, the new stacks are forced to be consistent with the anchor configuration.

Recall from the definition of loops in Section 2.1 that all unpaired bases can be uniquely assigned to a loop. Additionally, a stack does not interleave with the anchor configuration if and only if it is defined on unpaired bases within a single loop. This forms the basis of the final procedure RNASCF. See Figure 4.

Procedure COMPUTEANCHORCONFIGURATION(k, f)

1. Pick a pair of sequences (A, B) at random from the set \mathcal{R} of RNA sequences.
2. Compute putative stacks \mathcal{P}^A from A , and \mathcal{P}^B from B with minimum length k . k is chosen according to the lengths of sequences in \mathcal{R} , and typically $k = 4$.
3. Compute the optimum configuration. Reduce \mathcal{P}^A to retain only the stacks from the optimum configuration. Denote \mathcal{P}'^A as the reduced set.
4. For each sequence $R \in \mathcal{R}$, compute the optimum pairwise configuration of (A, R) using the reduced set \mathcal{P}'^A . Denote $M[(A, B), \mathcal{R}]$ as the sum of the configuration costs.
5. Recompute Steps 1-4 for various random choices of (A, B) , and pick the pair (A, B) with minimum configuration cost $M[(A, B), \mathcal{R}]$.
6. Retain only the stacks in \mathcal{P}'^A that appear in $1 - f$ fraction of the sequences in \mathcal{R} . Denote the subset as \mathcal{P}''^A . Output \mathcal{P}''^A as the *anchored structure* of \mathcal{R} .

Fig. 3. Procedure for computing anchor configuration.

Procedure RNASCF(k, f)

1. $\mathcal{P} = \text{ComputeAnchorConfiguration}(k, f)$
2. In each sequence, partition the unpaired bases according to their loop region in \mathcal{P} .
3. For every loop region that has a minimum number of unpaired bases, predict additional putative stacks with $k' < k$. Each 'arm' of the stack is constrained to have contiguous base-pairs.
4. For each stack in the optimal configuration that was not present in every member of the family, recompute the alignment with the additional putative stacks to retrieve less conserved stacks.
5. For each set of loop regions and potential stacks, recurse using $\text{RNASCF}(k', f)$ to compute additional stacks in the loop regions.

Fig. 4. The procedure RNASCF for computing Consensus Folds.

3.3 Implementation Details

The program is implemented in C, and is available upon request. In default setting, we limit ourselves to two iterations. For the first iteration, we choose the default parameters as $k = 4, h = 8$ (k is the minimum length and h is the minimum number of hydrogen bonds in the putative stacks) and $g = 0$ (no unpaired bases allowed in stacks). For the second iteration, the default settings are changed to $k = 3, h = 6$.

4 Results and Discussion

To test the performance of `RNAscF`, we chose a set of 12 RNA families from the Rfam database [33]. 20 sequences were chosen for each family, except for CRE (RF00220) and glmS (RF00234) for which we chose available 10 sequences respectively. Stacks were retrieved from the annotated structures for each of these

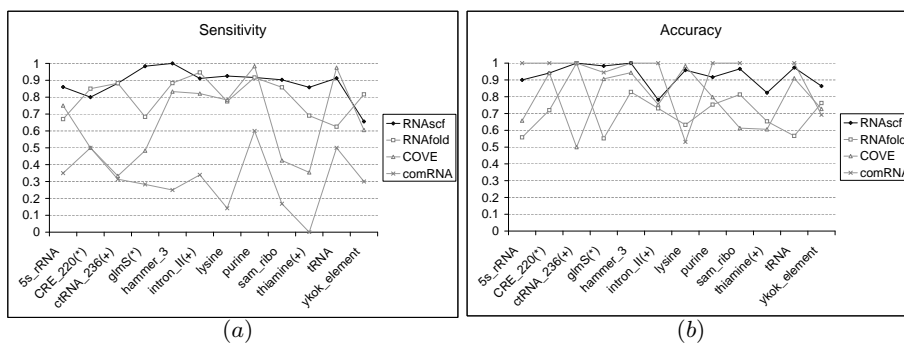


Fig. 5. Sensitivity and accuracy of RNA secondary structure prediction on 12 RNA families. The default parameters for **RNAscF** are $f=0.7$ and $k=4$ and $h=8$ for the first iteration. **RNAfold** is run under default parameters. **COVE** is run under the default parameters, using the multiple alignment from **ClustalW** as input. **comRNA** is run under the recommended parameter ($p=0.7$, $s=0.56$). (*) There are only 10 sequences in these families. (+) **RNAscF** is run under $k=3$ and $h=6$ on these families, due to their small size.

sequences. In all, there are 953 stacks. We chose 3 other programs to compare the performance of **RNAscF**, choosing the best representative of different methodologies: **RNAfold**, which is an implementation of energy based minimization [15] from the Vienna package; **COVE**, is an implementation of covariance model [23], and **comRNA** [29], which is based on computing anchors in multiple sequence alignment. Only **comRNA** predicts stacks explicitly. **COVE** and **RNAfold** do not explicitly predict stacks, but most of their base-pairs appear in stacks. For best results, we ignored unstacked base-pairs (with unpaired bases on either side) for **RNAfold** and **COVE**. Larger stack length cutoffs were also tried, but this choice gave the best balance of sensitivity and accuracy. *Sensitivity* is defined as the fraction of true stacks that overlapped with predicted stacks. A sensitivity of 1 would imply that all true stacks overlapped with some predicted stacks. Correspondingly *accuracy* is the fraction of predicted stacks that overlapped with a true stack. As **COVE** expects aligned sequences, we aligned the sequences using **ClustalW** [36]. The alignment was used to train the Covariance model, and the model was then used to align sequences, and predict structure. We also ran **COVE** on unaligned sequences, but the performance in that case was inferior to the performance on **ClustalW** aligned sequences.

Figure 5 shows the plots of the sensitivity and specificity of all programs on the test (detailed numbers are shown in Table 1). As can be seen in the tables, **RNAscF** is at the top or near the top in every family, and maintains high sensitivity and accuracy throughout, with an average accuracy of 0.884, and average sensitivity of 0.926. Only **comRNA** shows consistently high accuracy because it predicts very few stacks (leading to low-sensitivity) that are well-conserved in both sequence and structure. **COVE** occasionally shows very poor

sensitivity, possibly because of incorrect seed alignment. `RNAfold` predicts many stacks, and therefore has good sensitivity, but the extra predictions lead to a loss of accuracy. While our method shows robust performance for a limited number of given RNA sequences, its performance improves when the number of the given sequences increase.

Table 1. A complete list of the comparison of sensitivity and accuracy of RNA secondary structure prediction on 12 RNA families shown in Figure 5.

Name (Rfam_id)	Sensitivity				Accuracy			
	RNAscf	RNAfold	Cove	comRNA	RNAscf	RNAfold	Cove	comRNA
5s_rRNA (RF00001)	0.86	0.67	0.75	0.35	0.9	0.558	0.658	1
Rhino_CRE (RF00220)	0.8	0.85	0.5	0.5	0.941	0.72	0.941	1
ctRNA_pGAI (RF00236)	0.882	0.882	0.333	0.313	1	1	0.5	1
glmS(RF00234)	0.983	0.683	0.483	0.283	0.983	0.552	0.906	0.944
Hammerhead_3 (RF00008)	1	0.883	0.833	0.25	1	0.828	0.943	1
Intron_gpII (RF00029)	0.91	0.946	0.821	0.339	0.782	0.731	0.754	1
Lysine (RF00168)	0.925	0.775	0.783	0.142	0.958	0.633	0.984	0.531
Purine (RF00167)	0.917	0.917	0.983	0.6	0.917	0.753	0.797	1
Sam_riboswitch (RF00162)	0.903	0.858	0.425	0.168	0.966	0.813	0.613	1
Thiamine (RF00059)	0.858	0.690	0.354	0	0.824	0.654	0.606	-
tRNA (RF00005)	0.912	0.625	0.975	0.5	0.973	0.567	0.910	1
ykok (RF00380)	0.656	0.817	0.606	0.3	0.863	0.762	0.727	0.692
Average	0.884	0.8	0.654	0.312	0.926	0.714	0.778	0.924

Finally, we emphasize that even though sometimes we cannot predict all the stacks in all the given sequences, the consensus structure obtained by `RNAscf` is always the right configuration; the prediction errors in a few input sequences are usually due to an incorrect stack that is very close to a correct one. Since this cannot be quantified, we use one example to demonstrate that the minor prediction error in a few given sequences does not affect the prediction of the common structure. Figure 6 shows the predicted configuration of the four programs on the SAM riboswitch. Clearly, `RNAfold` can predict the correct configuration in some of the RNA sequences, but make the wrong prediction on the others. This is not surprising, because it analyzes each RNA sequence separately and doesn't presume they have the common structure. However, it will be difficult to derive their common structure based on these results. `comRNA` tends to miss many real stacks, although the ones it predicts are often correct. `COVE` predicted some correct stacks but it may miss some correct stacks and also predict some wrong ones. Similar results were seen for all families.

In conclusion, `RNAscf` establishes the principle that anchored stacks selection based on seed configurations, and prediction of consensus structure subject to anchored constraints is a valid approach to RNA structure prediction. Our future work will be aimed at correcting errors by using a stochastic iterative scheme such as Gibbs sampling [37]. In each step, we will remove a stack from the consensus structure, and add a new stack sampled from possibilities that are consistent with the remaining configuration, and weighted according to the energy. Early experiments have shown the promise of such refinement. An expended version of this paper is available upon request.

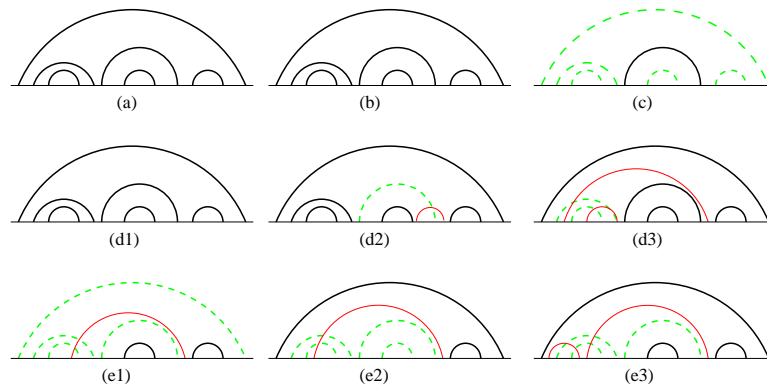


Fig. 6. A comparison of predicted stack configurations by different programs. (a) The true consensus stacks configuration for the sam riboswitch (RF00162). (b) *RNAscf* prediction. (c) *comRNA* prediction. (d1)-(d3) The first three *RNAfold* predictions. (e1)-(e3) The first three *COVE* predictions. Note that *RNAfold* and *COVE* are not limited to predicting conserved stacks, and therefore give potentially a different answer for each sequence. Thick line, dashed (green) line and thin (red) line represent true stacks, missed stacks and wrong stacks in the corresponding predicted configurations.

References

1. Eddy, S.: Non-coding RNA genes and the modern RNA world. *Nat Rev Genet* **2** (2001) 919–929
2. Storz, G.: An expanding universe of noncoding RNAs. *Science* **296** (2002) 1260–63
3. International Human Genome Sequencing Consortium: Finishing the euchromatic sequence of the human genome. *Nature* **431** (2004) 931–945
4. Kampa, D., et al.: Novel RNAs identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. *Genome Res* **14** (2004) 331–342
5. Nahvi, A., Sudarshan, N., Ebert, M., Zou, X., Brown, K., Breaker, R.: Genetic control by a metabolite binding mRNA. *Chemical Biology* **9** (2003) 1043–1049
6. Vitreschak, A., et al.: Riboswitches: the oldest mechanism for the regulation of gene expression? *Trends in Genetics* **20** (2003) 44–50
7. Tinoco, I., Uhlenbeck, O., Levine, M.: Estimation of secondary structure in ribonucleic acids. *Nature* **230** (1971) 362–367
8. Nussinov, R., Jacobson, A.: Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc Natl Acad Sci U S A* **77** (1980) 6309–6313
9. Nussinov, R., Pieczenik, G., Griggs, J., Kleitman, D.: Algorithms for loop matchings. *SIAM J. Appl. Math.* **35** (1978) 68–82
10. Smith, T., Waterman, M.: RNA Secondary structure. *Math. Biosci.* **42** (1978) 257–266
11. Waterman, M.: Secondary structure of single stranded nucleic acids. *Adv. Math. Suppl. Stud.* **I** (1978) 167–212
12. Zuker, M., Sankoff, D.: RNA secondary structure and their prediction. *Bull. Math. Biol.* **46** (1984) 591–621
13. Zuker, M.: Prediction of RNA secondary structure by energy minimization. *Methods Mol Biol* **25** (1994) 267–294
14. Hofacker, I.: Vienna RNA secondary structure server. *Nucl Acids Res* **31** (2003) 3429–3431

15. Jaeger, J., Turner, D., Zuker, M.: Improved predictions of secondary structures for RNA. *Proc Natl Acad Sci U S A* **86** (1989) 7706–7710
16. Pavesi, G., Mauri, G., Stefani, M., Pesole, G.: RNAProfile: an algorithm for finding conserved secondary structure motifs in unaligned RNA sequences. *Nucl Acids Res* **32** (2004) 3258–3269
17. Levitt, M.: Detailed molecular model for transfer ribonucleic acid. *Nature* **224** (1969) 759–763
18. Hofacker, I., Fekete, M., Stadler, P.: Secondary structure prediction for aligned RNA sequences. *J Mol Biol* **319** (2002) 1059–1066
19. Gorodkin, J., Stricklin, S., Stormo, G.: Discovering common stem-loop motifs in unaligned RNA sequences. *Nucl Acids Res* **29** (2001) 2135–2144
20. Sankoff, D.: Simultaneous solution of the RNA folding, alignment and protosequence problems. *Siam J. Appl. Math.* **45** (1985) 810–825
21. Mathews, D., Turner, D.: Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J Mol Biol* **317** (2002) 191–203
22. Gorodkin, J., Heyer, L., Stormo, G.: Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucl Acids Res* **25** (1997) 3724–32
23. Eddy, S., Durbin, R.: RNA sequence analysis using covariance models. *Nucl Acids Res* **22** (1994) 2079–2088
24. Sakakibara, Y., et al.: Recent methods for RNA modeling using Stochastic Context Free Grammars. *Combinatorial Pattern Matching* **807** (1994)
25. Knudsen, B., Hein, J.: Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucl Acids Res* **31** (2003) 3423–3428
26. Knight, R., Birmingham, A., Yarus, M.: BayesFold: rational 2 degrees folds that combine thermodynamic, covariation, and chemical data for aligned RNA sequences. *RNA* **10** (2004) 1323–1336
27. Bray, N., Pachter, L.: MAVID: Constrained Ancestral Alignment of Multiple Sequences. *Genome Res.* **14** (2004) 693–699
28. Waterman, M.: Consensus methods for folding single-stranded nucleic acids. *Mathematical methods for DNA Sequences* (1989) 185–224
29. Ji, Y., Xu, X., Stormo, G.: A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics* **20** (2004) 1591–1602
30. Perriquet, O., Touzet, H., Dauchet, M.: Finding the common structure shared by two homologous RNAs. *Bioinformatics* **19** (2003) 108–116
31. Bouthinon, D., Soldano, H.: A new method to predict the consensus secondary structure of a set of unaligned RNA sequences. *Bioinformatics* **15** (1999) 785–798
32. Davydov, E., Batzoglou, S.: A computational model for rna multiple structural alignment. *Combinatorial Pattern Matching* (2004)
33. Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A., Eddy, S.: Rfam: an RNA family database. *Nucl Acids Res* **31** (2003) 439–441
34. Touzet, H., Perriquet, O.: CARNAC: folding families of related RNAs. *Nucl Acids Res* **32** (2004) 142–145
35. Bafna, V., Muthukrishnan, S., Ravi, R.: Computing similarity between RNA strings. *Combinatorial Pattern Matching* **937** (1995) 1–14
36. Thompson, J., Higgins, D., Gibson, T.: 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** (1994) 4673–4680
37. Lawrence, C., et al.: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262** (1993) 208–214