Structural Alignment of RNAs Using Profile-csHMMs and Its Application to RNA Homology Search: Overview and New Results

Byung-Jun Yoon, *Member, IEEE,* and P. P. Vaidyanathan, *Fellow, IEEE*

***Abstract*— Systematic research on noncoding RNAs (ncRNAs) has revealed that many ncRNAs are actively involved in various biological networks. Therefore, in order to fully understand the mechanisms of these networks, it is crucial to understand the roles of ncRNAs. Unfortunately, the annotation of ncRNA genes that give rise to functional RNA molecules has begun only recently, and it is far from being complete. Considering the huge amount of genome sequence data, we need efficient computational methods for finding ncRNA genes. One effective way of finding ncRNA genes is to look for regions that are similar to known ncRNA genes. As many ncRNAs have well-conserved secondary structures, we need statistical models that can represent such structures for this purpose. In this paper, we propose a new method for representing RNA sequence profiles and finding structural alignment of RNAs, based on profile context-sensitive HMMs (profile-csHMMs). Unlike existing models, the proposed approach can handle any kind of RNA secondary structures, including pseudoknots. We show that profile-csHMMs can provide an effective framework for the computational analysis of RNAs and the identification of ncRNA genes.**

***Index Terms*— Noncoding RNA (ncRNA) gene prediction, profile context-sensitive HMM (profile-csHMM), sequential component adjoining (SCA) algorithm, RNA similarity search.**

I. Introduction

The various cellular mechanisms that sustain the life of living organisms are carried out by the elaborate collaborations of numerous biomolecules, such as DNA, RNA, and proteins. For a long time, proteins have been believed to be the most important molecules among them, which perform most of the structural, catalytic, and regulatory roles in all cells. In the meanwhile, DNA has been mainly viewed as the reservatory for protein coding information and RNAs have been regarded as passive intermediary molecules that simply interconnect DNA and proteins.

However, a number of recent observations in molecular biology indicate that this traditional view may have been too restrictive and incomplete to explain many biological functions in complex multicellular organisms, such as plants, insects, and animals. Recent studies on various genomes has revealed that there are numerous noncoding RNAs (ncRNAs), which are RNA molecules that are not translated into proteins but directly function as RNAs, that play crucial roles in various biological processes [7], [16], [25]. In addition to the well-known examples such as tRNAs (transfer RNAs) and rRNAs (ribosomal RNAs), functional ncRNAs have been found to be abundant, and the functions of the ncRNAs that have been identified till now are truly diverse. For example, ncRNAs are known to be involved in gene silencing [24], RNA processing [31], RNA modification [23], translation and transcription control [29], just to name a few.

As RNAs can directly interact with other RNA and DNA molecules in a sequence-specific manner, they can be especially useful in regulatory mechanisms, where the recognition of a specific nucleotide sequence is required [7]. In fact, it has been shown that many ncRNAs are actively involved in controlling various gene regulatory networks [12]. Examples of such regulatory RNAs include miRNAs (microRNAs) [2], riboregulators [9], and riboswitches [26]. In higher organisms such as mammals, the genomic output seems to be dominated by ncRNA transcripts, which suggests that the greater portion of their genome may be dedicated to regulating the development of cells [16].

Based on these observations, it becomes clear that we cannot fully understand the precise mechanisms of various biological networks, unless we first understand the roles of ncRNAs in these networks. Unfortunately, the annotation of ncRNA genes, which are regions in the DNA that give rise to functional ncRNAs, has begun only recently, and it is still far from being complete [17]. Although several systematic screenings of various genomes have identified many ncRNAs, it is believed that there still exist numerous ncRNAs that have not been discovered yet. Given the enormous amount of genome sequence data that is still growing at a fast pace, finding ncRNA genes solely by experimental means is practically infeasible. For a fast annotation of ncRNA genes in genome sequences, it is crucial to develop efficient computational methods for finding these genes.

One effective method for finding new ncRNA genes is to search for regions that look similar to known ncRNA genes. This is typically called a *similarity search* or a *homology search.* As many ncRNAs have secondary structures that are well-conserved among different species, it is important to incorporate this structural information in the search. In fact, scoring schemes that effectively combine contributions from the sequence similarity and the structural similarity are known to be much more discriminative than schemes that are based on sequence similarity alone [8].

Until now, a number of statistical models have been pro-

Manuscript received January 13, 2007; received August 29, 2007. This work was supported in part by the NSF grant CCF-0636799.

Byung-Jun Yoon is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA.

P. P. Vaidyanathan is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, USA.

posed that can be used for representing RNA secondary structures and implementing scoring schemes that combine sequence similarity and structural similarity [5], [21], [15]. However, these models can handle only a limited class of RNA secondary structures. For example, the CMs (covariance models) [6], which have been widely used in RNA sequence analysis, and the PHMMTSs (pair hidden Markov models on tree structures) [21], which are a more recent development, cannot handle RNAs that have pseudoknots.[1] As there exist many RNAs with functionally important pseudoknots [13], [27], this can be potentially a serious limitation. Recently, another method has been proposed based on PSTAGs (pair stochastic tree adjoining grammars) [15] that can handle many known pseudoknots, but not all of them.

In this paper, we propose a new method for representing RNA sequence profiles and building RNA sequence analysis tools. The proposed method is based on profile context-sensitive hidden Markov models (profile-csHMMs) [38], and it can in principle handle any kind of pseudoknots. To demonstrate the effectiveness of the new approach, we build a structural alignment tool for RNAs, which can be directly used for computing the similarity score between two RNAs. Experimental results will show that the profile-csHMM based approach can achieve high prediction ratios at a relatively low computational cost, providing an effective framework for building tools for finding ncRNA genes.

A. Scope and Outline

The purpose of this paper is twofold. We first present an overview of recent results on context-sensitive HMMs (csHMMs) and profile-csHMMs, and then we present new results on the application of profile-csHMMs in RNA sequence analysis. The paper is organized as follows. In Sec.II, we begin with a review of RNA and RNA secondary structure, and we give a brief overview of RNA similarity search. In Sec.III, we review the concept of csHMMs that has been proposed in [32], [37]. Context-sensitive HMMs are extensions of traditional HMMs that can effectively describe long-range correlations between distant symbols, and they have been shown to be useful in RNA sequence analysis [33], [34], [39]. In Sec.IV, we elaborate on profile-csHMMs [38], which are a subclass of csHMMs that are especially useful in representing RNA sequence profiles. A dynamic programming algorithm that can be used for finding the optimal path in a profile-csHMM is described in Sec.V. In Sec.VI, we propose a new method for finding structural alignments of RNAs based on profile-csHMMs. Experimental results of the proposed method are presented in Sec.VII and Sec.VIII, where it is compared to other existing methods. The paper is concluded in Sec.IX.

II. Review of Statistical Models for Representing RNA Sequences

A. RNA Secondary Structure

RNA is a nucleic acid polymer that consists of four types of nucleotides. The nucleotides are denoted by A, C, G, and U, which stand for *adenine*, *cytosine*, *guanine*, and *uracil*. In DNA, uracil is replaced by *thymine* (T), and they are chemically similar to each other. A-U and C-G can form hydrogen-bonded base-pairs, which are called Watson-Crick base-pairs. In addition to the canonical A-U and C-G pairs, non-canonical pairs do also exist, where the most common non-canonical pair is the G-U wobble base-pair. Bases that can form a base-pair are typically said to be complementary to each other. Unlike DNA, which exists in a double-stranded form (called DNA double helix), RNA molecules are generally single-stranded.

[1]RNA secondary structures that have crossing base-pairs are called pseudoknots. Formal definition of a pseudoknot can be found in Sec.II.

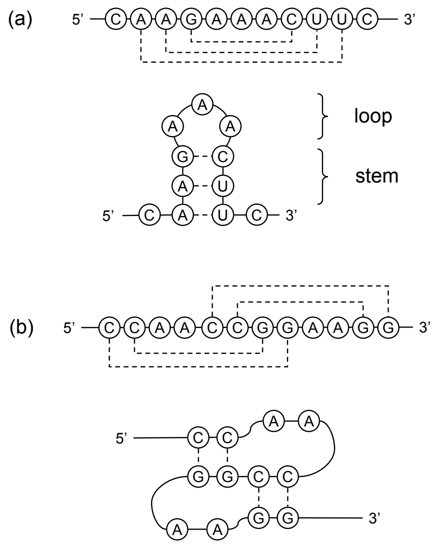


Fig. 1. Examples of RNA secondary structures. The dashed lines indicate the interactions between bases that form complementary base-pairs. (a) RNA with a hairpin (stem-loop) structure. (b) RNA with pseudoknots.

Due to the interactions between the complementary bases, an RNA molecule often folds onto itself to form a number of stacked base-pairs. The two-dimensional structure that results from this intramolecular folding is called the *RNA secondary structure*. In contrast, the one-dimensional nucleotide sequence is called the *primary sequence* of the RNA. Examples of RNA secondary structures are shown in Fig.1. For example, the RNA shown in Fig.1(a) forms three base-pairs after folding. These stacked base-pairs are called a *stem*. The unpaired bases that are bounded by the base-pairs are called a *loop*. For this reason, the secondary structure in Fig.1(a) is usually called a *stem-loop* structure (or a *hairpin* structure, due to its shape). Fig.1(b) shows another interesting example of an RNA secondary structure. Unlike the RNA in Fig.1(a), where all base-pairs occur in a nested manner, the RNA shown in Fig.1(b) has crossing base-pairs. To be more precise, let us consider a base-pair between the positions i and j ($i < j$), and another base-pair between the positions k and ℓ ($k < \ell$). If the base-pairs (i, j) and (k, ℓ) satisfy

$$i < k < \ell < j \quad \text{or} \quad k < i < j < \ell$$

we say that the two pairs are *nested*. On the other hand, if

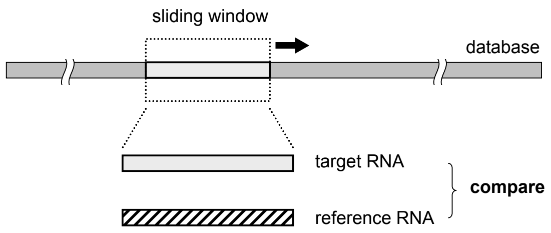


Fig. 2. Illustration of an RNA similarity search.

they satisfy

$$i < k < j < \ell \quad \text{or} \quad k < i < \ell < j$$

we say that these pairs are *crossing*. RNA secondary structures that have crossing base-pairs are typically called *pseudoknots*. In most cases, the base-pairs in an RNA secondary structure occur in a nested manner. However, there exist also many RNAs with pseudoknots [13], [27]. Crossing base-pairs introduce some complications in RNA sequence analysis, as will be shown later.

B. Scoring Scheme for Comparing RNA Sequences

For many ncRNAs, their secondary structures play pivotal roles in carrying out their biological functions. As a result, many RNA families have characteristic secondary structures that are commonly shared by their members [5]. Sometimes, this secondary structure can be still observed when there is little similarity between the primary sequences of two members. For this reason, it is important to consider both the primary sequence and the secondary structure when performing an RNA similarity search.

In an RNA similarity search, we scan a sequence database to look for regions that closely resemble the *reference RNA* that is of our interest. Generally, we use a sliding window and compare the *target RNA* that is located inside the window with the reference RNA. This is illustrated in Fig.2. Note that the size of the sliding window need not be fixed. In general, we use a variable window, with restrictions on its minimum and maximum lengths. If the target RNA is 'similar enough' to the reference RNA, we report it as a putative member that is likely to belong to the same family as the reference RNA. In order to decide whether the two RNAs are similar or not, we need a scoring scheme that can give us a quantitative measure of the similarity between them. As homologous RNAs conserve their primary sequences as well as their secondary structures, this scoring scheme should be able to reasonably combine the contributions from sequence similarity as well as structural similarity. In fact, it has been observed that such a combined scoring scheme can significantly enhance the specificity of an RNA similarity search [8].

Let us consider how we can devise such a scoring scheme. Measuring the similarity between two primary sequences is relatively straightforward. Conceptually, we can simply align the sequences and see where the two sequences differ from each other. We can assign negative scores for base substitutions

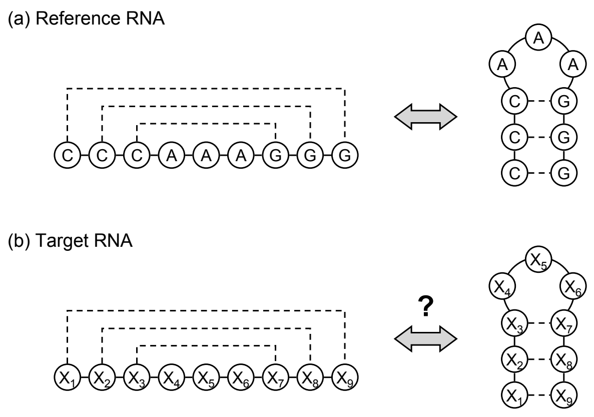


Fig. 3. Comparing the structural similarity between two RNAs. (a) A reference RNA with a stem-loop structure. (b) A target RNA with an unknown structure. We can investigate its base correlations to see whether the given RNA can fold to the same secondary structure as the reference RNA.

and gaps, while assigning positive scores for identical bases. For example, let us consider the following alignment.

$$\begin{array}{cccccc} A & C & C & G & - & U \\ - & C & G & G & A & U \end{array} \tag{1}$$

If we respectively assign -3, -4, and $+1$, for each base substitution, gap, and identity, the primary sequence similarity score for the alignment shown in (1) will be

$$-4 + 1 - 3 + 1 - 4 + 1 = -8.$$

Since there can be many different ways for aligning the two sequences, one immediate question is which alignment should be used for computing the similarity score. A reasonable and widely used solution is to find the optimal alignment that maximizes the score, and use this maximum score as a quantitative measure of their similarity. There are efficient algorithms for finding the optimal alignment [5].

However, it is not immediately obvious how we can compare the secondary structures of two RNAs. Given a reference RNA with a specific secondary structure, how can we figure out whether this structure is conserved in the target RNA? In order to answer this question, let us consider the example in Fig.3(a). The reference RNA shown in Fig.3(a) has a stem-loop structure. As mentioned earlier, this secondary structure results from the interactions (shown in dashed lines) between the complementary bases that make the RNA molecule fold onto itself. Now, let us consider a target RNA with an unknown structure, as shown in Fig.3(b). If this RNA is to fold to the same structure as the reference RNA, what kind of conditions should be satisfied by its primary sequence? From Fig.3(b), we can easily see that in order for this to be true, the bases (X_1, X_9), (X_2, X_8), and (X_3, X_7) should form complementary base-pairs. For example, 'GAACACUUC' and 'ACGAAACGU' can fold to the same secondary structure, while 'CCCAAAUUU' cannot.

This shows that we can represent an RNA secondary structure in terms of base correlations in the primary sequence of the RNA. Therefore, in order to develop a scoring scheme

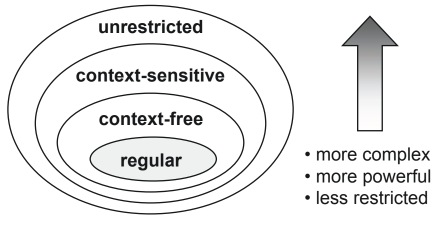


Fig. 4. Chomsky hierarchy of transformational grammars.

that can properly combine the contributions from sequence similarity and structural similarity between a reference RNA and a target RNA, we need a statistical model that can effectively represent the base correlations in the reference RNA.

C. Modeling RNA Secondary Structures

Then, what kind of statistical models can we use for modeling the base correlations that arise from a conserved RNA secondary structure? We can find the answer by examining the so-called *Chomsky hierarchy of transformational grammars* [3]. A transformational grammar can be viewed as a set of 'symbol rewriting rules (production rules)' that can be repetitively used to generate a set of symbol sequences over a given alphabet. Chomsky categorized transformational grammars into four classes, namely, *regular grammars*, *context-free grammars*, *context-sensitive grammars*, and *unrestricted grammars*, in the order of increasing descriptive power. The Chomsky hierarchy is illustrated in Fig.4.

An RNA with a secondary structure contains one or more *symmetric* regions (or more precisely, *reverse complementary* regions) in the primary sequence, due to the complementary base-pairs that make the RNA fold. In this sense, we can view RNAs with conserved secondary structures as *biological palindromes*. Palindromes are symmetric sequences that read the same in either direction. Due to the symmetry, palindromes have strong correlations between distant symbols.

It is known that the regular grammars, which are the simplest among the four classes in the Chomsky hierarchy, cannot describe palindrome languages [3], [5]. The HMMs (hidden Markov models), which have been widely used in various applications, can be viewed as *stochastic regular grammars*, hence they cannot be used for describing palindrome languages. It is of course possible that a HMM generates palindromes, but the important point is that we cannot construct a HMM that generates *only* such palindromes. For this reason, HMMs cannot effectively discriminate between palindromes and non-palindromes, which makes them unsuitable for developing an RNA scoring scheme.

In order to model palindrome languages, we have to use higher-order grammars, such as the context-free grammars. Context-free grammars can effectively describe symbol correlations that occur in a nested manner. As most RNA secondary structures have nested base-pairs, SCFGs (stochastic context-free grammars) have been extensively used in RNA sequence analysis [5], [8].[2] However, context-free grammars are inherently incapable of describing crossing correlations. As a result, SCFGs cannot handle RNA pseudoknots, which can be potentially a serious limitation. In order to overcome this problem, two subclasses of context-sensitive grammars (CSGs) have been proposed relatively recently [15], [18]. These grammars can handle a large number of known pseudoknots, but neither of them can handle all pseudoknots.

Instead of using these grammars, we can use the context-sensitive HMMs that have been recently proposed [32], [37]. As we will show in the following section, csHMMs can describe *any* kind of pairwise symbol correlations, hence they are capable of handling any kind of RNA secondary structures, including pseudoknots. For further discussions on RNA sequence analysis, the reader is referred to [5], [8], [40].

III. Context-Sensitive Hidden Markov Models

Context-sensitive HMMs are extensions of conventional HMMs and they have been recently introduced in [32], [37]. In a csHMM, certain states have variable emission and transition probabilities that depend on the 'context'. Emissions made at specific states are stored in the memory, and this data (or the context) is used to adjust the probabilities of some future states. This context-dependent property is very useful in modeling long-range correlations between distant symbols, and it can significantly increase the descriptive power of HMMs.

Unlike conventional HMMs, csHMMs have three different kinds of hidden states, namely, *single-emission states* S_n, *pairwise-emission states* P_n, and *context-sensitive states* C_n. Single-emission states are identical to regular states in traditional HMMs, and they have fixed emission probabilities that do not depend on the context. Pairwise-emission states are similar to single-emission states in the sense that their emission probabilities are also fixed. The difference is that the symbols emitted at pairwise-emission states are stored in the associated memory[3], so that they can be used to adjust the probabilities at the context-sensitive states. When we enter a context-sensitive state, it first accesses the associated memory to retrieve the symbol that has been previously emitted at the corresponding pairwise-emission state. The emission probabilities of the context-sensitive state is adjusted according to the retrieved symbol. As the pairwise-emission state P_n and the context-sensitive state C_n work cooperatively, they always exist in pairs, where each state pair is assigned a separate memory. As we need a context to adjust the emission probabilities at a context-sensitive state, the transition probabilities in the model are adjusted (based on the status of the memory that is associated with the context-sensitive state), such that we cannot enter a context-sensitive state when the memory is empty.

By arranging the pairwise-emission states and the corresponding context-sensitive states appropriately, we can represent any kind of pairwise symbol correlations. For example,

[2]Note that the CM (covariance model) is a SCFG with a special structure.

[3]The associated memory can be a stack or a queue, depending on the type of correlations we want to model. For modeling RNA secondary structures, it is usually more convenient to use a stack.

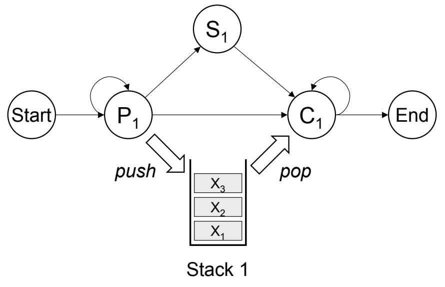


Fig. 5. A context-sensitive HMM that generates only palindromes.

using a csHMM, we can easily construct a model that generates *only* palindromes. An example of such a model is shown in Fig.5. As we can see in Fig.5, the csHMM has a single-emission state S_1, a pairwise-emission state P_1, and a context-sensitive state C_1. In this example, the state pair (P_1, C_1) uses a stack. Initially, the model begins at the pairwise-emission state P_1. It can make several self-transitions to emit a number of symbols, which will be stored in the stack. At some point, the model will enter the context-sensitive state C_1. When we enter C_1, we retrieve a symbol from the top of the stack, and the emission probabilities of C_1 are adjusted such that it emits the same symbol as the retrieved one. Transition probabilities of C_1 are adjusted such that it makes self-transitions until the stack becomes empty. Once the stack is empty, the model terminates. In this way, the csHMM in Fig.5 can generate palindromes that take one of the following forms

$$\begin{aligned} \mathbf{x}_1 &= x_1x_2\ldots x_N x_N \ldots x_2 x_1 && \text{(even length)} \\ \mathbf{x}_2 &= x_1x_2\ldots x_N x_{N+1} x_N \ldots x_2 x_1 && \text{(odd length).} \end{aligned}$$

The underlying state sequences for $\mathbf{x}_1$ and $\mathbf{x}_2$ will be

$$\mathbf{y}_1 = \underbrace{P_1 \ldots P_1}_{N \text{ states}} \underbrace{C_1 \ldots C_1}_{N \text{ states}} \quad \text{and} \quad \mathbf{y}_2 = \underbrace{P_1 \ldots P_1}_{N \text{ states}} S_1 \underbrace{C_1 \ldots C_1}_{N \text{ states}},$$

respectively. In case we want to generate biological palindromes that are reverse complementary to themselves, we can simply adjust the emission probabilities of C_1 such that it emits bases that are complementary to the bases emitted at P_1. By adjusting the context-sensitive emission probabilities at C_1, we can model any kind of base-pairs including non-canonical pairs.

Similarly, using a csHMM, we can also construct a model that generates only symbol sequences of the form

$$\mathbf{x}_3 = x_1x_2\ldots x_N x_1 x_2 \ldots x_N,$$

which is a concatenation of two identical sequences. A language that contains only such sequences is called a *copy language*. Fig.6(a) shows an example of a csHMM that represents a copy language. Note that the csHMM in Fig.6(a) uses a queue instead of a stack. An interesting thing about a copy language is that it gives rise to symbol correlations that cross each other. This can be clearly seen in Fig.6(b). As we have mentioned earlier, such correlations cannot be described by SCFGs, needless to mention HMMs.

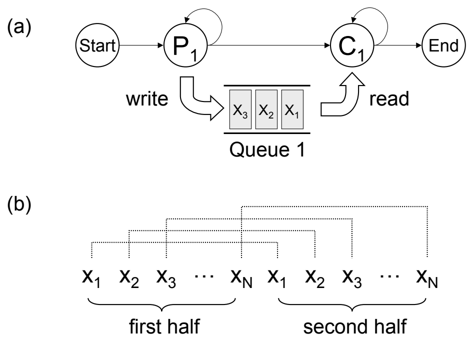


Fig. 6. (a) An example of a csHMM that represents a copy language. (b) Symbol sequences in a copy language contain crossing correlations.

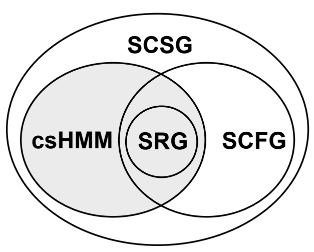


Fig. 7. The position of csHMMs in the Chomsky hierarchy.

It would be interesting to find out where the csHMM lies in the Chomsky hierarchy. This is illustrated in the Venn diagram shown in Fig.7. As context-sensitive HMMs are generalizations of conventional HMMs, it completely contains the SRGs (stochastic regular grammars). The csHMMs have a considerable overlap with SCFGs (stochastic context-free grammars), but neither of them fully contain the other [37]. For example, there are many languages that include sequences with *crossing* correlations, which can be represented by a csHMM but not by a SCFG. One such example is the copy language that can be modeled by the csHMM shown in Fig.6(a). Similarly, there exist languages that can be described by a SCFG but not by a csHMM. Such an example can be found in [37].[4]

There exist efficient algorithms for csHMMs that can be used for finding the optimal state sequence of an observed symbol sequence [35], [37], computing the probability of the symbol sequence [36], [37], and for optimizing the model parameters [37]. It has been shown that csHMMs can be effectively used in RNA sequence analysis [33], [34], [39]. For further discussions on csHMMs and their applications in computational RNA sequence analysis, the reader is referred to these references.

IV. Profile Context-Sensitive HMM

In the previous section, we have seen that csHMMs can easily model correlations between non-adjacent symbols by

[4]In practice, as far as representing RNA secondary structures is concerned, any structure that can be represented by a SCFG is also representable by a csHMM.

arranging the pairwise-emission states and the corresponding context-sensitive states in an appropriate manner. This is indeed very convenient for describing the base correlations that are frequently observed in RNA sequences, and we can use csHMMs to represent various kinds of RNA secondary structures [32]. In this section, we review the concept of *profile context-sensitive HMMs (profile-csHMMs)* [38], which are a subclass of csHMMs with a special structure. Profile-csHMMs are especially useful for building probabilistic sequence profiles of RNA families, as we show next.

A. Representing Consensus RNA Sequences

Let us assume that we are given a multiple alignment of RNA sequences that belong to the same RNA family. How can we construct a model that can statistically represent the common patterns and the important motifs in this alignment? We typically call such a representation a *probabilistic sequence profile* or a *consensus sequence* of the RNA family. Once we have constructed this model, it can be used for scoring new RNA sequences and finding homologous RNAs.

One model that has been widely used for building probabilistic profiles of protein coding genes and protein sequences is the profile-HMM [5]. Profile-HMMs are a subclass of HMMs that have linear repetitive structures (i.e., state transition diagrams). Because of their convenience in modeling sequence profiles, many coding-gene finders have been built based on profile-HMMs [5]. Since HMMs are incapable of describing the base correlations that arise from RNA secondary structures, profile-HMMs cannot be directly used for representing consensus RNA sequences. However, we can construct csHMMs in a similar manner so that they become suitable for representing RNA sequence alignments. Such csHMMs are called profile-csHMMs [38].

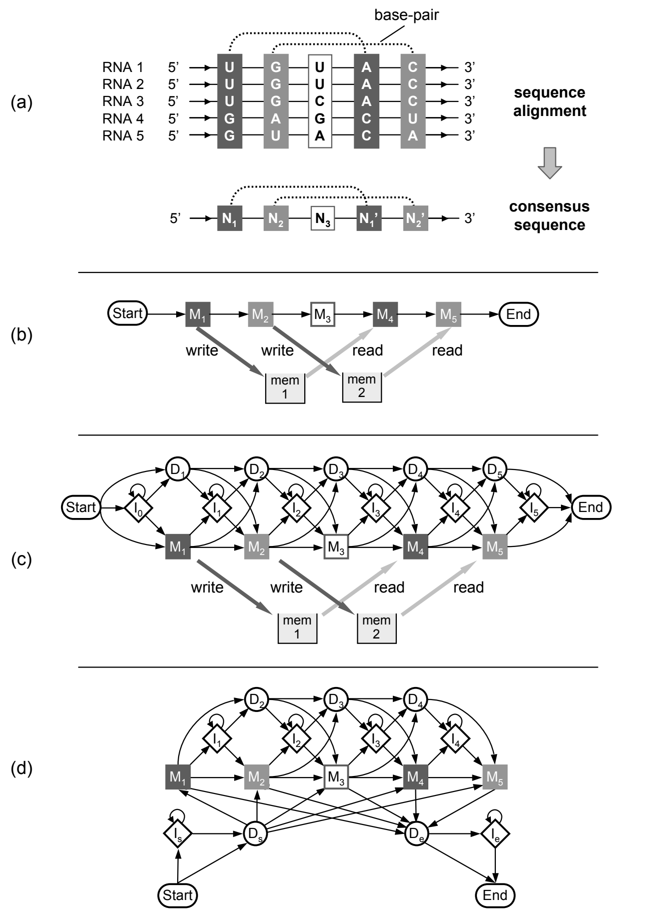


Fig. 8. Constructing a profile-csHMM from an RNA sequence alignment. (a) Example of an RNA sequence alignment. The consensus RNA has two base-pairs. (b) An ungapped profile-csHMM that corresponds to the consensus RNA sequence. (c) The final profile-csHMM that allows additional insertions and deletions in the consensus sequence. (d) A variant of the standard profile-csHMM that allows local alignment.

B. Constructing a Profile-csHMM

The structure of a profile-csHMM is similar to that of a conventional profile-HMM. Profile-csHMMs repetitively use three kinds of states, namely, *match states* M_k, *insert states* I_k, and *delete states* D_k.

1) Building an ungapped model: The match state M_k is used to represent the case when a base in the observed RNA sequence matches the k-th base in the consensus RNA sequence that was used to construct the profile-csHMM. For this reason, the number of match states in a profile-csHMM is identical to the length of the consensus RNA sequence. Each match state M_k can have a different set of emission probabilities, so that we can describe the observed frequencies of the four bases at each position.

The main difference between the conventional profile-HMMs and the profile-csHMMs is that the profile-csHMMs have three different types of match states. As we have seen in Sec.III, csHMMs have three distinct types of states, which are single-emission states, pairwise-emission states, and context-sensitive states. Each M_k can choose from these three types, hence there will be *single-emission match states*, *pairwise-emission match states*, and *context-sensitive match states*. Single-emission match states are used to represent the base positions that are not involved in base-pairing. For two positions that form a base-pair, we use a pair of pairwise-emission match state and the corresponding context-sensitive match state to describe the correlation between these bases.

As an example, let us consider the RNA sequence alignment shown in Fig.8(a). Since the length of the consensus RNA sequence is five, we need five match states to represent the sequence. As we can see in Fig.8(a), the secondary structure of the consensus RNA has two base-pairs. In order to model the base-pair between the first and the fourth bases, we use a pairwise-emission state for M_1 and the corresponding context-sensitive state for M_4. Similarly, we use a pairwise-emission state for M_2 and a context-sensitive state for M_5. As the third base does not form a base-pair, we simply use a single-emission state for M_3. By interconnecting the match states $M_1, M_2, \ldots, M_5$, we obtain an *ungapped profile-csHMM* as shown in Fig.8(b). The ungapped model serves as the 'backbone' of the final profile-csHMM, and it can represent RNA sequences that match the consensus RNA sequence without any gap.

2) Modeling additional insertions and deletions: After constructing the ungapped model, we can add insert states I_k and delete states D_k to obtain the final profile-csHMM. These states are used to represent insertions and deletions in the observed RNA sequence. For example, let us consider the case when the observed RNA is longer than the consensus RNA. In this case, if we align the two RNAs, there will be one or more bases in the observed RNA that are not present in the original RNA. Such bases are modeled by insert states. The insert state I_k is used to handle insertions between the positions k and $k+1$ in the consensus RNA sequence. As the inserted bases are not correlated to other bases, we use single-emission states for insert states.[5] Now, let us consider the case when the observed RNA is shorter than the consensus RNA. If we align these RNAs, there will be one or more bases that are present in the consensus RNA but are missing in the observed RNA. Such deletions can be handled by delete states, where the state D_k is used to model the deletion of the k-th symbol in the original RNA. As delete states deal with 'missing' bases, these states are *non-emitting states*, which are simply used to interconnect other states. The final profile-csHMM that corresponds to the RNA sequence alignment in Fig.8(a) is shown in Fig.8(c).

3) Allowing Local Alignments: Although the profile-csHMM shown in Fig.8(c) assumes that the observed target RNAs will be globally aligned to the model, it is quite straightforward to change the model to allow local alignments as well. For example, we can simply allow transitions from the START state to any I_k or M_k states, and similarly, allow transitions from any I_k or M_k states to the END state. In this way, we can handle sequences that match the model only locally. Another way to allow local alignments is to use the structure shown in Fig.8(d), which is similar to that of the profile-HMM variant used in the *HMMER* package [5]. The states I_s and I_e are used to model the flanking sequences at the beginning and the end of the original sequence profile, respectively. D_s and D_e are non-emitting states, which are respectively used to allow transitions to (and from) any match state.

C. Descriptive Power of Profile-csHMMs

As we can see in the previous example, profile-csHMMs provide a simple and intuitive way of representing RNA sequence profiles. One important advantage of profile-csHMMs is their large descriptive power. In fact, profile-csHMMs can represent *any* kind of base-pair correlations by arranging the pairwise-emission match states and the context-sensitive match states in an appropriate manner. Therefore, profile-csHMMs are capable of representing any kind of pseudoknots unlike many existing models. As mentioned earlier, CMs (covariance models) [6] and PHMMTSs (pair hidden Markov models on tree structures) [21] can only represent RNA secondary structures with nested correlations, hence incapable of dealing with pseudoknots. PSTAGs [15] are capable of representing pseudoknots with exactly 2-crossing property. A secondary structure is said to have a m-crossing property, if there exist $m (\geq 2)$ base-pairs in the given secondary structure such that any two pairs in these m base-pairs cross each other. Fig.9(a) shows an example of an RNA secondary structure with 2-crossing property. PSTAGs can handle many known pseudoknots, as a large portion of known pseudoknots has 2-crossing property. But there also exist more complex pseudoknots that are beyond the descriptive power of PSTAGs. One such example is the flavivirus 3' UTR pseudoknot family [22], which will be considered in our experiments presented in Sec.VII. It will be shown that profile-csHMMs can be used for modeling and predicting the secondary structure of these pseudoknots. Profile-csHMM can also represent RNAs with even more complex secondary structures as those shown in Fig.9(b) (RNA with 3-crossing property) and Fig.9(c) (RNA with 4-crossing property), in a similar manner as described in Sec.IV-B.

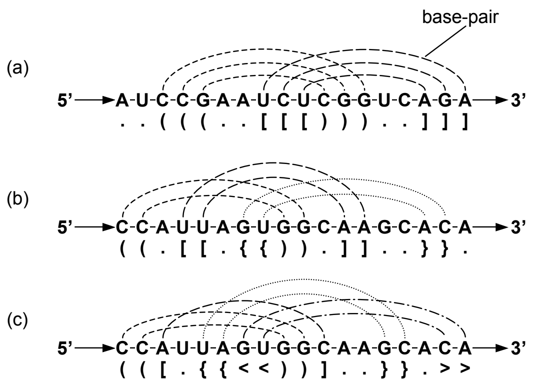

base-pair
(a)
5' A-U-C-C-G-A-A-U-C-U-C-G-G-U-C-A-G-A 3'
. . (((. . [[[ ))) . .]]]
(b)
5' C-C-A-U-U-A-G-U-G-G-C-A-A-G-C-A-C-A 3'
((. [[. { {)) .]] . . } } .
(c)
5' C-C-A-U-U-A-G-U-G-G-C-A-A-G-C-A-C-A 3'
(([. { { < <))] . . } } . > >

Fig. 9. Various types of RNA secondary structures. (a) RNA with 2-crossing property. (b) RNA with 3-crossing property. (c) RNA with 4-crossing property.

V. Computing the Optimal Alignment Score Based on Profile-csHMMs

Once we have constructed a profile-csHMM that statistically represents an RNA sequence family, this model can be used for finding new RNAs that look similar to the given RNA family. Let us assume that we are given a new RNA sequence (a 'target' RNA), and we want to find out how close it is to the RNA family under consideration (the 'reference' RNA). How can we compute the similarity score between the target RNA and the RNA family that is used as the reference? One good solution is to use the maximum observation probability of the target sequence based on the profile-csHMM that represents the reference RNA family. When using a csHMM, there can be many different state sequences (or *paths*) that give rise to the same symbol sequence, but each with a different probability. Therefore, in order to compute the maximum probability of an observed symbol sequence, we have to find the optimal path that maximizes the observation probability. As this can be viewed as finding the best alignment between a symbol

[5]In principle, we can also allow insertions of additional base-pairs. This can be done by using a pair of a pairwise-emission insert state and a context-sensitive insert state.

sequence and the given model, it is typically called the *optimal alignment* problem. Since the number of paths increases exponentially with the length of the observed sequence, we need an efficient algorithm for finding the optimal path in a systematic way. When using traditional HMMs, we can utilize the Viterbi algorithm [28] for finding the optimal path and computing the maximum observation probability. For SCFGs, we can use the CYK (Cocke-Younger-Kasami) algorithm [5] for this purpose.

However, as profile-csHMMs can describe many complicated symbol correlations that cannot be described by HMMs nor SCFGs, we need a more general algorithm that can deal with such correlations. Although we cannot directly use the existing algorithms, we can generalize them to develop an optimal alignment algorithm for csHMMs. In this section, we describe a dynamic programming algorithm called the *SCA (sequential component adjoining) algorithm* that can be used for the optimal alignment of profile-csHMMs. The basic idea of the SCA algorithm has been proposed in [38]. In the following, we describe the algorithm in more detail, with additional discussions on several important issues, such as the adjoining order of the algorithm and the computational complexity for handling RNAs with various secondary structures.

A. Two Generalizations

The SCA algorithm iteratively finds the optimal state sequence in a similar way as the Viterbi algorithm and the CYK algorithm. Given an observation sequence, the SCA algorithm first finds the optimal state sequences of short subsequences, and then it uses this information to find the optimal state sequences of longer subsequences. This process is repeated until we find the optimal state sequence of the entire observed sequence. In order to handle more complex correlations, the SCA algorithm makes the following generalizations to the existing algorithms.

Firstly, instead of using a subsequence that has a single interval, the SCA algorithm can use a subsequence that consists of multiple non-overlapping intervals. In order to define a subsequence, the SCA algorithm uses an ordered set of variable number of closed intervals. Let us consider an observed symbol sequence $\mathbf{x} = x_1 x_2 \dots x_L$. We define a set of I non-overlapping intervals $\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_I\}$, where $\mathbf{n}_k = [n_k^\ell, n_k^r]$ denotes the interval $n_k^\ell \le n \le n_k^r$. The subscript k indicates the k-th interval, and the superscripts ℓ and r are used to designate the 'left' and the 'right' ends of the given interval, respectively. We assume that the intervals in $\mathcal{N}$ are ordered, such that they satisfy

$$n_i^r < n_j^\ell \quad \text{for} \quad i < j. \tag{2}$$

Based on the set $\mathcal{N}$, the subsequence $\mathbf{x}(\mathcal{N})$ is defined as follows

$$\mathbf{x}(\mathcal{N}) = \underbrace{x_{n_1^\ell} \dots x_{n_1^r}}_{\mathbf{n}_1} \ \underbrace{x_{n_2^\ell} \dots x_{n_2^r}}_{\mathbf{n}_2} \ \dots \ \underbrace{x_{n_I^\ell} \dots x_{n_I^r}}_{\mathbf{n}_I}.$$

Note that $\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_I\}$ is generally *not* a partition of the entire interval $[1, L]$, hence the subsequence $\mathbf{x}(\mathcal{N})$ usually contains only a portion of the observation sequence $\mathbf{x}$.

Using this notation, we can define subsequences of $\mathbf{x}$ as follows

$$\begin{aligned}
\mathcal{N}_1 = \{[2,4]\} \quad &\Longrightarrow \quad \mathbf{x}(\mathcal{N}_1) = x_2 x_3 x_4 \\
\mathcal{N}_2 = \{[1,2],[4,6]\} \quad &\Longrightarrow \quad \mathbf{x}(\mathcal{N}_2) = x_1 x_2 \ x_4 x_5 x_6 \\
\mathcal{N}_3 = \{[1,1],[3,4],[7,8]\} \quad &\Longrightarrow \quad \mathbf{x}(\mathcal{N}_3) = x_1 \ x_3 x_4 \ x_7 x_8.
\end{aligned}$$

This generalization considerably increases the number of ways in which the intermediate subsequences (used during the iterative process of finding the optimal state sequence) can be defined, extended, and adjoined. As we continue these iterations, the set $\mathcal{N}$ that defines the intermediate subsequence will approach a partition of the entire range $[1, L]$, ultimately covering the entire sequence.

Secondly, the SCA algorithm allows us to explicitly define how the optimal state sequences of shorter subsequences can be extended and adjoined to find the optimal state sequences of longer subsequences. When using the SCA algorithm, there can be numerous ways to define the intermediate subsequences. These intermediate subsequences have to be defined in such a way that can take care of all the correlations in the profile-csHMM. Therefore, we cannot find the optimal state sequence of the observed sequence simply by proceeding left-to-right (as the Viterbi algorithm) or by proceeding inside-to-outside (as the CYK algorithm). In fact, we have to define a model-dependent *adjoining order* that specifies how we should define the intermediate subsequences, and how the extension and adjoining rules should be applied. This is elaborated in Sec.V-E in more detail.

B. Notations

Before describing the details of the SCA algorithm, let us first define the notations. As before, we denote the observed symbol sequence as $\mathbf{x} = x_1 x_2 \dots x_L$. The underlying state sequence is denoted as $\mathbf{y} = y_1 y_2 \dots y_{\bar{L}}$. Note that the length $\bar{L}$ of the state sequence can be larger than the length L of the observation, since $\mathbf{y}$ can have one or more non-emitting states (i.e., delete states D_k). We assume that the length of the profile-csHMM (defined as the the number of match states in the model) is K. The emission probability of a symbol x at a single-emission state or a pairwise-emission state v is denoted by $e(x|v)$. The emission probability of a symbol x_c at a context-sensitive state w is denoted by $e(x_c|w, x_p)$, where x_p is the symbol that was previously emitted at the corresponding pairwise-emission state. The transition probability from state v to state w is denoted by $t(v, w)$.

Consider an ordered set of I intervals $\mathcal{N} = \{\mathbf{n}_1, \dots, \mathbf{n}_I\}$, where the intervals satisfy (2). For this set $\mathcal{N}$, we define $\mathcal{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_I\}$ to be an ordered set of state-pairs $\mathbf{s}_i = (s_i^\ell, s_i^r)$, where s_i^ℓ and s_i^r represent the hidden states at the left and right ends of the i-th interval $\mathbf{n}_i = [n_i^\ell, n_i^r]$. Now, we define $P(\mathbf{x}(\mathcal{N}), \mathbf{y}(\mathcal{N}))$ as the observation probability of the subsequence $\mathbf{x}(\mathcal{N})$, whose underlying state sequence is

$$\mathbf{y}(\mathcal{N}) = \underbrace{y_{\bar{n}_1^\ell} \dots y_{\bar{n}_1^r}}_{\mathbf{n}_1} \ \underbrace{y_{\bar{n}_2^\ell} \dots y_{\bar{n}_2^r}}_{\mathbf{n}_2} \ \dots \ \underbrace{y_{\bar{n}_I^\ell} \dots y_{\bar{n}_I^r}}_{\mathbf{n}_I}.$$

Since $\mathbf{y}(\mathcal{N})$ can contain non-emitting states, $\bar{n}_i^\ell$ may not satisfy $\bar{n}_i^\ell = n_i^\ell$. Similarly, we may have $\bar{n}_i^r \neq n_i^r$. Based on this notation, we finally define $\alpha(\mathcal{N},\mathcal{S})$ to be the *maximum log-probability* of the subsequence $\mathbf{x}(\mathcal{N})$

$$\alpha(\mathcal{N},\mathcal{S}) = \max_{\mathbf{y}(\mathcal{N})} \Big[\log P\left(\mathbf{x}(\mathcal{N}),\mathbf{y}(\mathcal{N})\right) \Big],$$

over all possible state sequences $\mathbf{y}(\mathcal{N})$ that satisfy $y_{\bar{n}_i^\ell} = s_i^\ell$ and $y_{\bar{n}_i^r} = s_i^r$ for all $i = 1,\ldots,I$. In addition to this, we define two variables $\lambda_a(\mathcal{N},\mathcal{S})$ and $\lambda_b(\mathcal{N},\mathcal{S})$ that will be used for tracing back the optimal state sequence that maximizes the observation probability.

C. Initialization

Initially, we begin with computing the optimal log-probability $\alpha(\mathcal{N},\mathcal{S})$ of subsequences that either consist of a single base or a single base-pair.

(i) For a position k $(1 \le k \le K)$ in the profile-csHMM where M_k is a single-emission match state, we let $\mathcal{N} = \{[n,n]\}$, $\mathcal{S} = \{(M_k,M_k)\}$ and initialize

$$\alpha(\mathcal{N},\mathcal{S}) = \log e(x_n|M_k)$$
$$\lambda_a(\mathcal{N},\mathcal{S}) = (\varnothing,\varnothing),\ \lambda_b(\mathcal{N},\mathcal{S}) = (\varnothing,\varnothing)$$

for all positions $1 \le n \le L$. Similarly, we let $\mathcal{N} = \{[n,n-1]\}$, $\mathcal{S} = \{(D_k,D_k)\}$ and initialize

$$\alpha(\mathcal{N},\mathcal{S}) = 0$$
$$\lambda_a(\mathcal{N},\mathcal{S}) = (\varnothing,\varnothing),\ \lambda_b(\mathcal{N},\mathcal{S}) = (\varnothing,\varnothing)$$

for all $1 \le n \le L+1$.

(ii) For positions j and k $(1 \le j < k \le K)$ where M_j is a pairwise-emission state and M_k is the corresponding context-sensitive state, we let $\mathcal{N}_1 = \{[n,n],[m,m]\}$, $\mathcal{S}_1 = \{(M_j,M_j),(M_k,M_k)\}$ and compute

$$\alpha(\mathcal{N}_1,\mathcal{S}_1) = \log e(x_n|M_j) + \log e(x_m|M_k,x_n)$$
$$\lambda_a(\mathcal{N}_1,\mathcal{S}_1) = (\varnothing,\varnothing),\ \lambda_b(\mathcal{N}_1,\mathcal{S}_1) = (\varnothing,\varnothing)$$

for all positions $1 \le n < m \le L$. Furthermore, we let $\mathcal{N}_2 = \{[n,n-1],[m,m-1]\}$, $\mathcal{S}_2 = \{(D_j,D_j),(D_k,D_k)\}$, and initialize the log-probability as follows

$$\alpha(\mathcal{N}_2,\mathcal{S}_2) = 0$$
$$\lambda_a(\mathcal{N}_2,\mathcal{S}_2) = (\varnothing,\varnothing),\ \lambda_b(\mathcal{N}_2,\mathcal{S}_2) = (\varnothing,\varnothing)$$

for all n and m $(1 \le n \le m \le L+1)$.

(iii) For single bases emitted at insert states, we let $\mathcal{N} = \{[n,n]\}$, $\mathcal{S} = \{(I_k,I_k)\}$, and initialize

$$\alpha(\mathcal{N},\mathcal{S}) = \log e(x_n|I_k)$$
$$\lambda_a(\mathcal{N},\mathcal{S}) = (\varnothing,\varnothing),\ \lambda_b(\mathcal{N},\mathcal{S}) = (\varnothing,\varnothing)$$

for all $0 \le k \le K$ and $1 \le n \le L$.

D. Adjoining Subsequences

After computing the optimal log-probabilities of all subsequences that consist of a single base or a single base-pair, we recursively adjoin these subsequences to obtain the optimal log-probabilities of longer subsequences. This can be done by applying the following adjoining rules.

Rule 1 Consider the log-probabilities $\alpha(\mathcal{N}^a,\mathcal{S}^a)$ and $\alpha(\mathcal{N}^b,\mathcal{S}^b)$ of the two subsequences $\mathbf{x}(\mathcal{N}^a)$ and $\mathbf{x}(\mathcal{N}^b)$, where

$$\mathcal{N}^a = \{\mathbf{n}_1^a,\ldots,\mathbf{n}_{I_a}^a\},\ \mathcal{S}^a = \{\mathbf{s}_1^a,\ldots,\mathbf{s}_{I_a}^a\}$$
$$\mathcal{N}^b = \{\mathbf{n}_1^b,\ldots,\mathbf{n}_{I_b}^b\},\ \mathcal{S}^b = \{\mathbf{s}_1^b,\ldots,\mathbf{s}_{I_b}^b\}.$$

We assume that there is no overlap between the symbol sequences $\mathbf{x}(\mathcal{N}^a)$ and $\mathbf{x}(\mathcal{N}^b)$ nor between the underlying state sequences $\mathbf{y}(\mathcal{N}^a)$ and $\mathbf{y}(\mathcal{N}^b)$. In this case, we can compute the optimal log-probability of the longer subsequence $\mathbf{x}(\mathcal{N})$ as follows

$$\alpha(\mathcal{N},\mathcal{S}) = \alpha(\mathcal{N}^a,\mathcal{S}^a) + \alpha(\mathcal{N}^b,\mathcal{S}^b)$$
$$\lambda_a(\mathcal{N},\mathcal{S}) = (\mathcal{N}^a,\mathcal{S}^a),\ \lambda_b(\mathcal{N},\mathcal{S}) = (\mathcal{N}^b,\mathcal{S}^b).$$

The sets $\mathcal{N}$ and $\mathcal{S}$ are unions of the smaller sets

$$\begin{aligned} \mathcal{N} &= \mathcal{N}^a \cup \mathcal{N}^b = \{\mathbf{n}_1,\ldots,\mathbf{n}_I\} \\ \mathcal{S} &= \mathcal{S}^a \cup \mathcal{S}^b = \{\mathbf{s}_1,\ldots,\mathbf{s}_I\}, \end{aligned}$$

where $I = I_a + I_b$ and the intervals $\mathbf{n}_i$ are relabeled such that they satisfy (2) and $\mathbf{s}_i \in \mathcal{S}$ corresponds to $\mathbf{n}_i \in \mathcal{N}$. ■

Rule 2 Assume that there exist two intervals $\mathbf{n}_i,\mathbf{n}_{i+1} \in \mathcal{N}$ that satisfy $n_i^r + 1 = n_{i+1}^\ell$, which implies that the two intervals $[n_i^\ell,\ n_i^r]$ and $[n_{i+1}^\ell,\ n_{i+1}^r]$ are adjacent to each other. For simplicity, let us assume that $i = I-1$. In this case, we can combine the two intervals $\mathbf{n}_{I-1}$ and $\mathbf{n}_I$ to obtain a larger interval

$$\mathbf{n}'_{I-1} = [n_{I-1}^\ell,\ n_I^r] = \{n |\ n_{I-1}^\ell \le n \le\ n_I^r\}$$

where the corresponding state-pair is $\mathbf{s}'_{I-1} = (s_{I-1}^\ell,\ s_I^r)$. Now, the log-probability $\alpha(\mathcal{N}',\mathcal{S}')$ for

$$\mathcal{N}' = \{\mathbf{n}_1,\ldots,\mathbf{n}_{I-2},\mathbf{n}'_{I-1}\},\ \mathcal{S}' = \{\mathbf{s}_1,\ldots,\mathbf{s}_{I-2},\mathbf{s}'_{I-1}\}$$

can be computed as follows

$$\begin{aligned} \alpha(\mathcal{N}',\mathcal{S}') &= \max_{n_{I-1}^r} \left(\max_{s_{I-1}^r, s_I^\ell} \Big[\alpha(\mathcal{N},\mathcal{S}) + \log t(s_{I-1}^r, s_I^\ell) \Big] \right) \\ (n^*, s_r^*, s_\ell^*) &= \operatorname*{arg\,max}_{(n_{I-1}^r, s_{I-1}^r, s_I^\ell)} \Big[\alpha(\mathcal{N},\mathcal{S}) + \log t(s_{I-1}^r, s_I^\ell) \Big] \\ \mathcal{N}^* &= \big\{ \mathbf{n}_1,\ldots,\mathbf{n}_{I-2},[n_{I-1}^\ell, n^*],[n^*+1, n_I^r] \big\} \\ \mathcal{S}^* &= \big\{ \mathbf{s}_1,\ldots,\mathbf{s}_{I-2},(s_{I-1}^\ell, s_r^*),(s_\ell^*, s_I^r) \big\} \\ \lambda_a(\mathcal{N}',\mathcal{S}') &= (\mathcal{N}^*,\mathcal{S}^*),\ \ \lambda_b(\mathcal{N}',\mathcal{S}') \ = \ (\varnothing,\varnothing). \end{aligned}$$

For $i < I-1$, we can similarly combine the two adjacent intervals $\mathbf{n}_i$ and $\mathbf{n}_{i+1}$ to obtain the optimal log-probability $\alpha(\mathcal{N}',\mathcal{S}')$ for the updated sets $\mathcal{N}'$ and $\mathcal{S}'$. ■

For simplicity, we have described the adjoining process in two distinct steps, namely, (i) adjoining two non-overlapping subsequences and (ii) combining adjacent intervals in a single subsequence. In practice, we can often combine these rules and apply them at the same time. This can be more convenient than applying them one by one. For example, if we know the optimal log-probability of two adjacent subsequences, where each subsequence consists of a single interval, we can adjoin the two sequences and combine the two intervals to compute the optimal log-probability of a longer subsequence that has also a single interval.

E. Adjoining Order

As mentioned earlier, when using the SCA algorithm we have to specify the adjoining order, according to which the adjoining rules should be applied. This adjoining order can be obtained from the consensus RNA sequence that was used to construct the profile-csHMM. Based on the consensus sequence, we first find out how the bases and the base-pairs in the given sequence can be adjoined one by one to obtain the entire sequence. During this procedure, we try to minimize the number of intervals that is needed to describe the intermediate subsequences, as a larger number of intervals leads to a higher computational cost for adjoining the subsequences.

An example is shown in Fig.10(a), which illustrates how we can obtain the consensus sequence in Fig.8(a) by sequentially adjoining its base and base-pairs. Note that the numbers inside the squares in Fig.10(a) indicate the original base-positions. Fig.10(b) shows the portion of the profile-csHMM that corresponds to the respective RNA subsequence at each step of Fig.10(a). Some steps in Fig.10(a) are subdivided into multiple steps in Fig.10(b) for illustration. Following this adjoining order, we can ultimately compute the maximum log-probability of the target RNA sequence and find the optimal state sequence that maximizes the probability. At each step, we compute the maximum log-probability of every possible subsequence (of the observed target RNA), whose underlying state sequence matches the corresponding portion of the profile-csHMM as shown in Fig.10(b).

For every step in the adjoining order, we first compute the log-probability $\alpha(\mathcal{N},\mathcal{S})$ of those subsequences, whose terminal states (i.e., s_i^ℓ, s_i^r) do not contain insert states. For example, in STEP 1, we compute $\alpha(\mathcal{N}_1,\mathcal{S}_1)$ for $\mathcal{N}_1=\{[n,n]\}$, $\mathcal{S}_1=\{(M_3,M_3)\}$ and $\mathcal{N}_1=\{[n,n-1]\}$, $\mathcal{S}_1=\{(D_3,D_3)\}$ using the initialization rules described in Sec.V-C. Note that the states in $\mathcal{S}_1$ correspond to the third base position in the original consensus sequence. Similarly, we compute the log-probability $\alpha(\mathcal{N}_2,\mathcal{S}_2)$ in STEP 2 (and also $\alpha(\mathcal{N}_4,\mathcal{S}_4)$ in STEP 4) based on the base-pair initialization rules in Sec.V-C. At some steps, the optimal log-probability is obtained by combining the log-probabilities computed in the previous steps. For example, in STEP 3, we can compute $\alpha(\mathcal{N}_3,\mathcal{S}_3)$ for $\mathcal{N}_3=\{[n_1,n_1],[n_2^\ell,n_2^r]\}$, $\mathcal{S}_3=\{(M_1,M_1),(M_3,M_4)\}$ by combining $\alpha(\mathcal{N}_1,\mathcal{S}_1)$ and $\alpha(\mathcal{N}_2,\mathcal{S}_2)$ as follows

$$\alpha(\mathcal{N}_3,\mathcal{S}_3)=\max_v\Big[\alpha(\mathcal{N}_1,\mathcal{S}_1)+\log t(v,M_4)+\alpha(\mathcal{N}_2,\mathcal{S}_2)\Big],$$

where $\mathcal{N}_1=\{[n_2^\ell,n_2^r-1]\}$, $\mathcal{S}_1=\{(M_3,v)\}$ and $\mathcal{N}_2=\{[n_1,n_1],[n_2^r,n_2^r]\}$, $\mathcal{S}_2=\{(M_1,M_1),(M_4,M_4)\}$. As shown in the above equation, we consider *all* possible transitions from state $v\in\{M_3,D_3,I_3\}$ to state M_4, and choose the one that maximizes the log-probability. In the example shown in Fig.10, the log-probability $\alpha(\mathcal{N}_1,\mathcal{S}_1)$ is computed in STEP 1 and $\alpha(\mathcal{N}_2,\mathcal{S}_2)$ is computed in STEP 2.

After computing these log-probabilities, we move on to compute the log-probabilities of those subsequences that have one or more insertions at the beginning and/or the end of some intervals. For example, at STEP 1, we can compute $\alpha(\mathcal{N}_1,\mathcal{S}_1)$ for $\mathcal{N}_1=\{[n,n+1]\}$ and $\mathcal{S}_1=\{(M_3,I_3)\}$ from

$$\alpha(\mathcal{N}_1,\mathcal{S}_1)=\alpha(\mathcal{N}_1^a,\mathcal{S}_1^a)+\log t(M_3,I_3)+\alpha(\mathcal{N}_1^b,\mathcal{S}_1^b),$$

where $\mathcal{N}_1^a=\{[n,n]\}$, $\mathcal{S}_1^a=\{(M_3,M_3)\}$, $\mathcal{N}_1^b=\{[n+1,n+1]\}$, $\mathcal{S}_1^b=\{(I_3,I_3)\}$. In a similar manner, we can also deal with a left insertion as well as multiple insertions.

F. Termination

By iteratively applying the adjoining rules, we can obtain the log-probability $\alpha(\mathcal{N},\mathcal{S})$ for $\mathcal{N}=\{[1,L]\}$ and $\mathcal{S}=\{(s^\ell,s^r)\}$, for all $s^\ell\in\{I_0,M_1,D_1\}$ and $s^r\in\{I_K,M_K,D_K\}$. Let us define $t(\text{START},s^\ell)$ to be the probability that the profile-csHMM will begin at the state s^ℓ, and $t(s^r,\text{END})$ as the probability that the model will terminate after s^r. The maximum log-probability of the entire observation sequence $\mathbf{x}$ can be computed as follows

$$\begin{aligned}\log P(\mathbf{x},\mathbf{y}^*) &= \max_{\mathbf{y}}\Big[\log P(\mathbf{x},\mathbf{y})\Big]\\ &= \max_{s^\ell,s^r}\Big[\log t(\text{START},s^\ell)\\ &\qquad +\alpha(\mathcal{N},\mathcal{S})+\log t(s^r,\text{END})\Big]\\ (s_\ell^*,s_r^*) &= \arg\max_{(s_1^\ell,s_1^r)}\Big[\log t(\text{START},s^\ell)\\ &\qquad +\alpha(\mathcal{N},\mathcal{S})+\log t(s^r,\text{END})\Big]\\ \lambda^* &= \Big([1,\ L],(s_\ell^*,s_r^*)\Big),\end{aligned}$$

where $\mathbf{y}^*$ is the optimal state sequence that maximizes the observation probability.

G. Trace-Back

Once we have computed the maximum log-probability $\log P(\mathbf{x},\mathbf{y}^*)$, we can trace-back the adjoining process to find the optimal state sequence $\mathbf{y}^*$ that gave rise to this log-probability. To describe the trace-back algorithm, let us define $\lambda_t=(\mathcal{N},\mathcal{S})$ and a stack T. The trace-back algorithm proceeds as follows.

1) Let $y_i=0$ $(i=1,2,\ldots,L)$.
2) Push λ^* onto the stack T.
3) Pop $\lambda_t=(\mathcal{N},\mathcal{S})$ from T. If $\lambda_t=(\varnothing,\varnothing)$, goto **step 6**. Otherwise, proceed to **step 4**.
4) If $\lambda_a(\lambda_t)\neq(\varnothing,\varnothing)$ push $\lambda_a(\lambda_t)$ onto T. Otherwise, $y_{n_i^\ell}=s_i^\ell$, for all $\mathbf{n}_i=[n_i^\ell,\ n_i^r]\in\mathcal{N}$ and the corresponding $\mathbf{s}_i=[s_i^\ell,\ s_i^r]\in\mathcal{S}$. (Note that when $\lambda_a(\lambda_t)=(\varnothing,\varnothing)$, we have $n_i^\ell=n_i^r$ and $s_i^\ell=s_i^r$.)
5) If $\lambda_b(\lambda_t)\neq(\varnothing,\varnothing)$ push $\lambda_b(\lambda_t)$ onto T.
6) If T is empty, proceed to **step 7**. Otherwise, goto **step 3**.

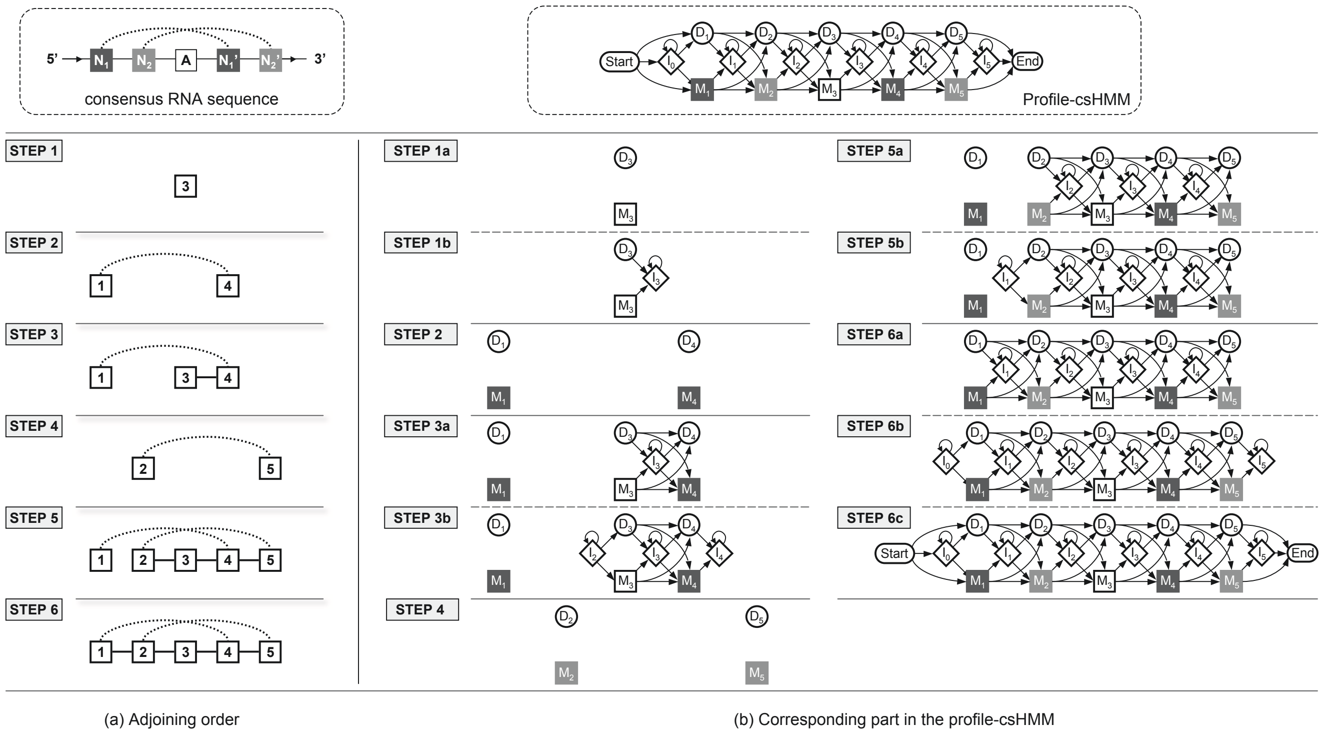


Fig. 10. (a) The adjoining order for the profile-csHMM shown in Fig.8(c). This shows how the reference RNA can be obtained by adjoining the base and base-pairs. (b) Corresponding parts in the profile-csHMM. This illustrates in which order the optimal state sequence can be found.

7) Let $\mathbf{y}^* = y_1 y_2 \dots y_{\bar{L}}$ and terminate.

At the end of the trace-back procedure, we can find the optimal state sequence $\mathbf{y}^*$ in the profile-csHMM that maximizes the observation probability of $\mathbf{x}$.

H. Principle of Optimality

In order to ensure that the "optimal state sequence" $\mathbf{y}^*$ obtained in Sec.V-G is indeed optimal, the following conditions should be satisfied. Firstly, when adjoining optimal subsequences (or combining intervals in an optimal subsequence), we have to make sure that (i) *all possible state transitions are considered* for every adjoining point, (ii) *all possible adjoining positions are compared* to each other, and that (iii) the *conditions that maximize the probability are chosen*. Secondly, the probabilities of the subsequences that are adjoined should be independent of each other. These two conditions ensure that all possible state sequences have been considered in finding the optimal probability of the new subsequence, and that there is no other partition of the given subsequence that will make the probability higher. Hence the new subsequence that is obtained by optimally adjoining shorter optimal subsequences will be also optimal. In fact, for a given profile-csHMM, if we define the adjoining order as in Fig.10(a) and proceed to find the optimal state sequence as described in Sec.V-E (and illustrated in Fig.10(b)), the aforementioned conditions are naturally satisfied. Therefore, at the end of the algorithm, it is guaranteed that the resulting state sequence $\mathbf{y}^*$ will be indeed optimal.

I. Computational Complexity of the SCA Algorithm

Unlike the Viterbi algorithm and the CYK algorithm, the computational complexity of the SCA algorithm is not fixed, and it depends on the adjoining order. As the adjoining order is specified based on the correlation structure of the profile-csHMM, the computational cost ultimately depends on the secondary structure of the consensus RNA sequence that was used to construct the model. For example, when we are dealing with an RNA that has a stem-loop structure, the complexity for finding the optimal alignment will be in the order of $O(L^2K)$, where K is the length of the profile-csHMM and L is the length of the target RNA. For this RNA, the SCA algorithm can simply proceed inside-to-outside (like the CYK algorithm) to find the optimal alignment. Therefore, its complexity is essentially identical to the complexity of using the CYK algorithm for parsing a CM without any "bifurcation rule" (used to generate multiple stems) [5]. Similarly, if the consensus RNA has multiple stems (without crossing correlations) like the tRNA cloverleaf structure, the complexity will be $O(L^3K)$. Again, this is identical to the complexity for using the CYK algorithm for parsing a general CM *with* bifurcation rules.

For RNAs with pseudoknots, which cannot be represented by SCFGs, the complexity becomes higher. In order to deal with crossing base correlations, we have to define intermediate subsequences with at least two intervals. As we have to consider all possible positions for the intermediate subsequences, there are $O(L^4)$ possibilities for choosing their positions. Furthermore, since the number of adjoining steps in the SCA algorithm is proportional to the length K of the profile-csHMM (which is identical to the length of the reference RNA) as illustrated in Fig.10, the computational complexity[6] of the SCA algorithm becomes at least $O(L^4K)$. As a comparison, note that the computational complexity of the PSTAG algorithm is also at least $O(L^4M)$, and it can be as large as $O(L^5M)$ depending on the structure of the PSTAG tree [15].

VI. Structural Alignment of RNAs Using Profile-csHMMs

As we have shown in the previous section, profile-csHMMs provide a convenient framework for statistically representing RNA sequence families, developing RNA similarity scoring schemes that can reasonably combine contributions from sequence similarity and structural similarity, and ultimately, building RNA homology search tools. To demonstrate the effectiveness of the proposed method, we have built a program that can be used for structural alignment of RNA sequences including pseudoknots.[7] Similar to the PSTAG-based alignment tool developed by Matsui *et al.* [15], it uses a single structured RNA sequence as a reference and aligns unfolded RNA sequences to it. However, unlike PSTAGs that can only handle pseudoknots with 2-crossing property, the given program can deal with a much larger class of RNAs.

The program proceeds as follows. It first constructs a profile-csHMM based on the reference RNA and its structural annotation. Instead of using a fully stochastic model with position-dependent emission probabilities, in this implementation, we have used the non-stochastic scoring matrix proposed in [11], as it has been used by several RNA analysis tools with good performance. Secondly, the program automatically finds the adjoining order that can be used for predicting the optimal state sequence of the profile-csHMM. The adjoining order is obtained in the following way. Let us consider a subsequence $\hat{\mathbf{x}}$ of the reference RNA that consists of $I\,(\leq I_{\max})$ intervals. We define $\Gamma(\hat{\mathbf{x}})$ as the number of base-pairs that are fully contained in the subsequence $\hat{\mathbf{x}}$. For a given $\hat{\mathbf{x}}$, we try to split it into two subsequences $\hat{\mathbf{x}}_\ell$ and $\hat{\mathbf{x}}_r$, where each of them may have up to $I_{\max}$ intervals, such that $\Gamma(\hat{\mathbf{x}}) = \Gamma(\hat{\mathbf{x}}_\ell)+\Gamma(\hat{\mathbf{x}}_r)$ is maximized. We begin this process by finding the best 'division' for the entire sequence, and proceed in a recursive manner until we reach the point where every subsequence consists of a single base or a single base-pair. The adjoining order of the profile-csHMM can be simply obtained by reversing this division process.

Currently, the program can deal with RNA secondary structures which can be handled by the SCA algorithm using subsequences with up to two intervals ($I_{\max} = 2$). This corresponds to the entire class of RNA secondary structures that can be represented by the grammar proposed by [18]. The so-called Rivas&Eddy class is regarded as the most general RNA class that is known today, and it covers almost all RNA secondary structures that have been identified until now [4]. Although the current implementation of the structural alignment program

[6]The complexity for aligning pseudoknots in the Rivas&Eddy class [18] will be at most $O(L^6K)$, which can be verified by a similar reasoning. For RNAs outside R&E class, the computational complexity can be even higher.

[7]The software and its C++ source code are available upon request.

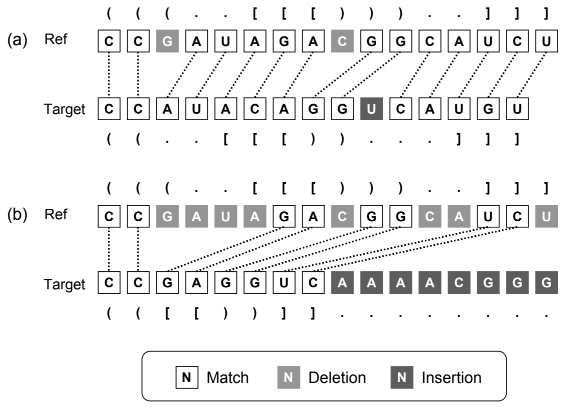


Fig. 11. Matching bases in an RNA sequence alignment. (a) The maximum distance between the matching bases is two. (b) The maximum distance between the matching bases is seven.

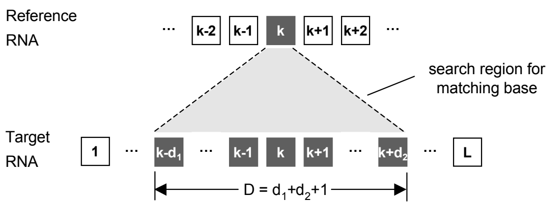


Fig. 12. Limiting the search region for finding the matching bases can significantly reduce the overall complexity of the structural alignment.

covers only the Rivas&Eddy class, it has to be noted that the capability of the profile-csHMMs and the SCA algorithm goes beyond the R&E class. For RNAs that are outside the R&E class, we can easily extend the current program to handle them.

One advantage of the given program is that it does not reject RNAs even if they are outside the descriptive capability of the current implementation. For example, if the reference RNA has a complex structure that is outside the R&E class, the program chooses the subset with maximum number of base-pairs such that the resulting secondary structure is contained in the R&E class.

Now, the constructed profile-csHMM can be used for carrying out a structural alignment between the reference RNA and a target RNA with unknown structure. We can follow the adjoining order that has been obtained in the previous step to find the optimal state sequence of the target RNA, which in turn yields the prediction of its secondary structure.

In implementing the SCA algorithm, we have introduced a parameter D, which is the length of the search region for finding the matching bases in the sequence alignment. This is motivated by the following observation. When we align two RNA sequences that are biologically relevant, the matching bases in the reference RNA and the target RNA are usually located very close to each other. Fig.11(a) shows an example of a typical sequence alignment, where the maximum distance between a base in the reference sequence and the matching base in the target sequence is two. Alignments with a large distance between the matching bases, as the one shown in Fig.11(b), are generally less probable. Based on this observation, we limit the search region as illustrated

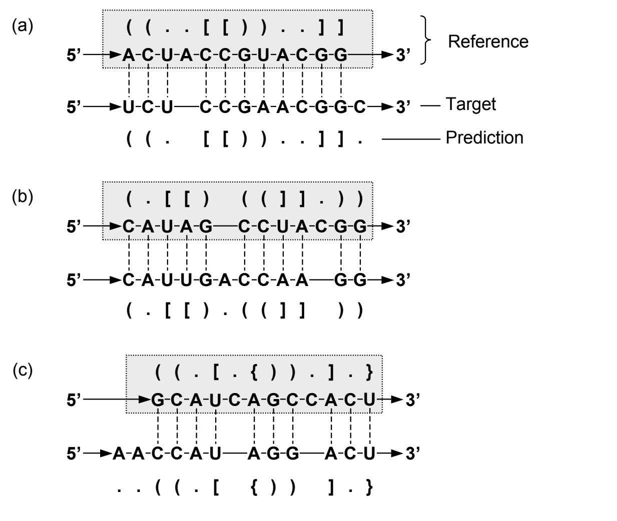


Fig. 13. Structural alignments of RNAs with various secondary structures.

in Fig.12. When looking for the base x_ℓ in the target RNA $\mathbf{x} = x_1 \dots x_L$ that matches the k-th base in the reference RNA, we only consider the bases between $\max(k - d_1, 1)$ and $\min(k + d_2, L)$, hence the maximum length of the search region is $D = d_1 + d_2 + 1$.

Restricting the search region has several advantages. First of all, it significantly reduces the overall complexity of the alignment algorithm, making the program practically usable in real applications. The computational complexity of aligning pseudoknots will be reduced from $O(L^4K)$ to $O(D^4K)$ (or from $O(L^6K)$ to $O(D^6K)$ in the worst case). For example, assume that we want to align two pseudoknots in the TOMBUS_3_IV RNA family (seed alignment) in the Rfam database [13]. The average length of these RNAs is around $L = 91$, and $D = 7$ is enough for finding the optimal alignment between any two members in the given family. In this case, limiting the search region reduces the overall complexity to around $(D/L)^4 \sim 0.35\%$ of the original. Secondly, when the structural alignment score obtained from the SCA algorithm is used for finding homologues of an RNA family, restricting the search region can yield better discrimination between homologues and non-homologues, as long as D is large enough to obtain the optimal alignment. As the optimal alignment of homologues is contained within the search space, the imposed restriction does not affect the alignment score of homologous sequences. However, limiting the search region will lead to an overall decrease in the alignment score of non-homologues, hence providing better discrimination between homologues and non-homologues.

A good way of choosing D is to compute the range between the matching bases in the original sequence alignment, and make it slightly larger than this range. Another method for estimating D is to construct a simple profile-HMM from the sequence alignment and find an (sequence-based) alignment between the target sequence and the profile-HMM. This alignment can be found very quickly, since the computational complexity for aligning a profile-HMM is only $O(LK)$. Then we compute the maximum distance between the matching bases in the given alignment, and use it to estimate D. This is indeed a very efficient strategy that results in a tremendous reduction in the CPU time needed for finding the structural alignments, while providing a good prediction performance, as will be demonstrated in Sec.VII.

TABLE I

PREDICTION PERFORMANCE OF PROFILE-CSHMM AND PSTAG.

	Profile-csHMM		PSTAG	
	SN (%)	SP (%)	SN (%)	SP (%)
CORONA_PK3	**96**	**97**	95	96
HDV_RIBOZYME	**95**	95	94	**96**
TOMBUS_3_IV	97	97	97	97
FLAVI_PK3	**95**	**96**	-	-

Fig.13 shows a few examples of structural alignments obtained from the program that has been just described. RNAs illustrated in Fig.13(a) and Fig.13(b) have 2-crossing properties and Fig.13(c) has a 3-crossing property. In each example, a target RNA with unknown structure is aligned to the reference RNA whose structure is known. As we can see in this example, the proposed approach can find good structural alignments for RNAs with various secondary structures.

VII. NUMERICAL EXPERIMENTS

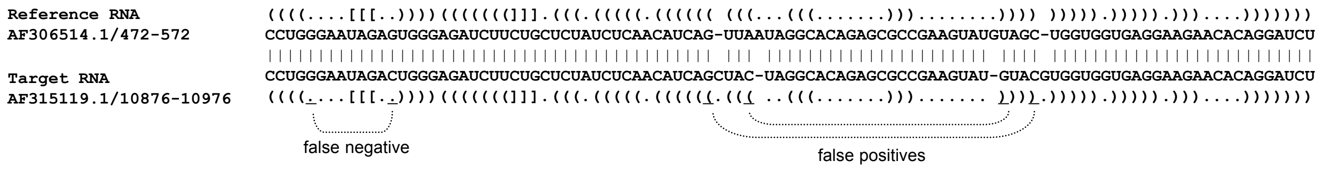
We tested the performance of our program using several pseudoknots included in the Rfam database [13]. The Rfam database provides a large collection of various RNA families, where the member sequences in each family are aligned to each other. In our experiments, we have used the sequences in the 'seed alignment' of each RNA family, as they are hand-curated and have reasonably reliable structural annotation. For each sequence family, we chose one of its members as the reference RNA, and used it along with its structural annotation to predict the secondary structure of all the other sequences in the same family. The predicted secondary structure has been compared to the annotated structure in the database, and we counted the number of correctly predicted base-pairs (true-positives; TP), the number of incorrectly predicted base-pairs (false-positives; FP), and the number of base-pairs in the annotated structure that were not predicted by the program (false-negatives; FN). These numbers have been used to compute the sensitivity (SN) and the specificity (SP) of the program that are defined as follows

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \mathrm{SP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}.$$

To obtain reliable estimates of these quantities, we performed a cross-validation experiment by repeating the previous process for every member in the given RNA family and computed the overall prediction ratios.

In order to compare the performance of the proposed method with that of PSTAGs, we first tested the program for three RNA families, CORONA_PK3, HDV_RIBOZYME, and TOMBUS_3_IV, which have all pseudoknot structures. Table I shows the prediction result of the proposed method along with the prediction result of PSTAGs.[8] In each case, the higher prediction ratio is boldfaced. As we can see in Table I,

[8]The prediction results of PSTAGs are obtained from [15] and have been rounded to integer values.



```
Reference RNA        ((((....[[[..))))(((((((]]].(((.(((((.((((( (((...(((.......)))........)))) ))))).))))).)))....)))))))
AF306514.1/472-572   CCUGGGAAUAGAGUGGGAGAUCUUCUGCUCUAUCUCAACAUCAG-UUAAUAGGCACAGAGCGCCGAAGUAUGUAGC-UGGUGGUGAGGAAGAACACAGGAUCU
                     |||||||||||||||||||||||||||||||||||||||||||| ||| |||||||||||||||||||||||| |||| ||||||||||||||||||||||||||
Target RNA           CCUGGGAAUAGACUGGGAGAUCUUCUGCUCUAUCUCAACAUCAGCUAC-UAGGCACAGAGCGCCGAAGUAU-GUACGUGGUGGUGAGGAAGAACACAGGAUCU
AF315119.1/10876-10976 ((((....[[[..))))(((((((]]].(((.(((((.((((((.((( ..(((.......))).......  )))).))))).))))).)))....)))))))
```

Fig. 14. Structural alignment of two RNAs in the FLAVI_PK3 family. The secondary structure of the target RNA has been predicted from the given alignment. Incorrect predictions (one false-negative and two false-positives) have been underlined.

TABLE II
AVERAGE CPU TIME FOR FINDING A STRUCTURAL ALIGNMENT.

	Average CPU Time (sec)	
	Profile-csHMM	PSTAG
CORONA_PK3	1.2	37.2
HDV_RIBOZYME	1.7	207.5
TOMBUS_3_IV	1.0	270.9
FLAVI_PK3	6.8	-

TABLE III
PREDICTION RESULTS OF THE PROPOSED METHOD USING RANDOMLY MUTATED RNA SEQUENCES.

	Profile-csHMM	
	SN (%)	SP (%)
CORONA_PK3	90	91
HDV_RIBOZYME	91	92
TOMBUS_3_IV	93	93
FLAVI_PK3	88	89

profile-csHMMs yielded accurate prediction results that are comparable to PSTAGs for all three RNAs that have been tested. But profile-csHMMs are more general than the PSTAGs as demonstrated in the next example.

Secondly, we tested the performance of the proposed method for the FLAVI_PK3 family that has a more complex secondary structure than the previous RNA families, which cannot be handled by the PSTAGs. The secondary structure of FLAVI_PK3 is similar to the example shown in Fig.13(b), which has two stems and additional base-pairs that cross the base-pairs in both stems. As we have already seen in Fig.13(b), profile-csHMMs are capable of dealing with such structures. Fig.14 shows a structural alignment of two RNAs in the FLAVI_PK3 family obtained using the proposed approach. Note that most base-pairs have been correctly predicted. There were two false-positives and a false-negative in the predicted structure when compared to the annotated structure in Rfam.

Despite the generality of the proposed method, its computational cost was much smaller than that of the PSTAGs. It can be seen that the profile-csHMM approach runs significantly faster than the PSTAG approach, despite its larger descriptive power. Table II shows the average CPU time that was needed for finding the structural alignment between two sequences in each RNA family.[9] In our experiments, the parameter D was automatically estimated by performing a simple sequence-based alignment as proposed in Sec.VI. It has to be noted that the initial alignment obtained by the profile-HMM is only used for estimating D, hence it does not affect the final structural alignment of the profile-csHMM.

In the preceding examples, there is considerable similarity between the primary RNA sequences in each category. However, the prediction performance of the proposed method does not depend strongly on the primary sequence similarity. The method is therefore applicable in cases where the sequences are related essentially by a common secondary structure alone. In order to show this, we randomly mutated the RNA sequences that were used in the previous experiments, such that the sequence similarity between the homologous RNAs got completely removed. During this process, bases that form complementary base-pairs were covaried to preserve the original secondary structure. The experimental results are summarized in Table III. As we can see in Table III, the prediction ratios have decreased only slightly, indicating that the proposed approach does not depend too much on sequence similarity.

[9]The experiments have been performed on a PowerMac G5 2.5 GHz with 4GB memory.

VIII. FURTHER COMPARISON

In Sec.VII, we compared the performance of the proposed method with that of PSTAG [15]. To the best of our knowledge, PSTAG is the first and the only grammar based method that can be used for representing RNA pseudoknots and finding their structural alignments. Therefore, it was most relevant to compare the profile-csHMM based structural alignment method with PSTAG. In this section, we provide further comparison with two other popular methods, which can be helpful in demonstrating the effectiveness of the proposed method.

A. Traditional Profile-HMM

Traditional profile-HMMs have been widely used for modeling protein sequences and protein-coding genes [5]. Profile-HMMs can be easily constructed based on multiple sequence alignments, and they are especially useful in performing similarity searches for proteins and coding genes. Given an observation sequence, traditional profile-HMMs find the optimal alignment between the model and the sequence solely based on *sequence similarity*. One interesting question would be how much we can improve the quality of the RNA alignments by using the *structural alignment* method based on profile-csHMMs, instead of using profile-HMMs. To verify the advantage of profile-csHMMs, we performed similar cross-validation experiments for predicting the structure of several RNA families, as in Sec.VII. For this experiment, we used sequences in the L20_LEADER, PURINE, and SRP_BACT

TABLE IV
PERFORMANCE COMPARISON BETWEEN PROFILE-CSHMM AND PROFILE-HMM.

	Profile-csHMM			Profile-HMM		
	SN (%)	SP (%)	Time (sec)	SN (%)	SP (%)	Time (sec)
L20_LEADER	84	84	2.97	61	67	2.72
PURINE	85	89	1.43	58	65	1.35
SRP_BACT	74	76	1.20	44	52	1.12

families in the Rfam database. Note that the average identity of the sequences in each of these families is relatively low.[10]

The structure prediction results of profile-csHMMs and traditional profile-HMMs are summarized in Table IV. The prediction accuracies of the profile-csHMM method were lower compared to the results in Table I, but still reasonably high. The main reason for the reduced accuracy is the relatively large structural variations among the members, rather than their low sequence similarity. As we have shown in Table III, the performance of the profile-csHMM based structural alignment method is not significantly affected by sequence similarity. In Table IV, we can also see that profile-csHMMs produced significantly better alignments than traditional profile-HMMs, whose prediction accuracies were around $20\% \sim 30\%$ higher. Nevertheless, the computational cost of the profile-csHMM approach was comparable to that of the profile-HMM approach.[11] In addition to the superior quality of the resulting alignments, another important advantage of the profile-csHMM approach is that it can yield good alignment scores that sensibly combine *structural similarity* and *sequence similarity*, which is crucial in RNA similarity search.

B. Iterated Loop Matching (ILM)

Recently, an efficient heuristic method, called *iterated loop matching* (ILM), has been proposed for predicting RNA secondary structures including pseudoknots [19]. ILM can utilize thermodynamic and/or comparative information to predict the secondary structure of individual RNAs or an alignment of RNAs, and it has been shown that ILM can achieve relatively high prediction accuracy at a low computational cost. However, ILM is mainly used for predicting RNA secondary structures, and it cannot be used for representing RNAs and finding structural alignments between structured and unstructured RNAs. Even though ILM is quite different from the proposed method in its goal and nature, comparing these methods might be useful in demonstrating the effectiveness of the profile-csHMM approach and showing the respective merits of the two methods. For this experiment, we again used the sequences in the FLAVI_PK3 and HDV_RIBOZYME families in Rfam. The prediction performance of the profile-csHMM method was measured based on cross-validation experiments as elaborated in Sec.VII. The performance of ILM has been measured by predicting the secondary structure of the individual sequences and comparing it to the annotated structure in the database. Predictions have been made using the ILM online server with the default parameters [20].

[10]The *average percentage identity* is typically used to measure the *sequence similarity* of the members in a given family. The average identities of L20_LEADER, PURINE, and SRP_BACT RNA families are 55%, 56%, and 50%, respectively [13].

[11]The average CPU time of the profile-csHMM approach summarized in Table IV consists of the time for estimating the search region (based on profile-HMM) and the time for finding the structural alignment using the SCA algorithm.

TABLE V
PERFORMANCE COMPARISON BETWEEN PROFILE-CSHMM AND ILM.

	Profile-csHMM		ILM	
	SN (%)	SP (%)	SN (%)	SP (%)
HDV_RIBOZYME	95	95	78	65
FLAVI_PK3	95	96	58	59

The prediction results of the two methods are summarized in Table V. From this table, we can see that the profile-csHMM based structural alignment method provides better performance than ILM, as it can exploit the structural annotation of the reference RNA. However, it has to be noted that the structural alignment method can only be used when we have a reference RNA whose structure is known. If we do not have a structured RNA that can be used as a reference, we have to use ILM or resort to other structure prediction methods.

IX. CONCLUDING REMARKS

In this paper, we proposed an effective framework for representing RNAs with various secondary structures including pseudoknots. The proposed approach was based on profile-csHMMs, which can be easily constructed from RNA multiple sequence alignments in a simple and intuitive manner. Experimental results indicate that the prediction accuracy of the profile-csHMM approach is comparable to the state-of-the-art. However, profile-csHMMs can handle a considerably larger class of secondary structures at a much lower computational cost.

The good prediction performance of the proposed scheme, as well as its generality and the relatively low computational cost makes profile-csHMMs an attractive choice for building homology search tools for noncoding RNAs. For example, we can build a family-specific prediction program similar to the tRNA CM [6], which finds new candidates that may belong to the given RNA family. Although we have used a non-stochastic scoring matrix in our program, we can also use a fully stochastic model with position-dependent probabilities to improve the specificity of the prediction program. These probabilities can be easily obtained from the multiple sequence alignment of the RNA family that is under consideration.

Another interesting application would be to build a BLAST-like tool that uses a single RNA with a known structure for finding structural homologues. Klein and Eddy [14] developed a database search program called RSEARCH that finds homologues of single structured RNAs, and they showed that it outperforms primary sequence based programs (including BLAST) in many cases. As RSEARCH is based on covariance models, they cannot be used for finding pseudoknots. We can develop a more general search program based on profile-csHMMs that can practically deal with any kind of RNA secondary structures.

Although the proposed approach can find structural alignments of RNAs in a relatively short time, it is still slow for scanning a large database. Recently, Weinberg and Ruzzo [30] suggested the use of heuristic profile-HMM filters to expedite CM-based searches. They showed that using such filters can make the scanning speed significantly faster at virtually no loss of performance. In a similar manner, it is possible to incorporate profile-HMM based pre-screening filters to speed-up the database search based on profile-csHMMs. We are currently investigating the optimal construction of such a pre-screening filter from a given profile-csHMM, and preliminary results indicate that considerable improvement in search speed can be achieved by incorporating this strategy [41].

Even though the main focus of this paper was on RNA similarity search, another useful approach for finding ncRNAs is the comparative sequence analysis. A common strategy of comparative methods is to find noncoding regions that are well-conserved among different species. Once we have identified such regions, they are investigated further to see whether they also share a common secondary structure, as this can be an indicator which shows that these regions correspond to functional RNAs. Further details on these methods can be found in [17].

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, "Basic local alignment search tool", *Journal of Molecular Biology*, vol. 215, pp. 403-410, 1990.

[2] D. P. Bartel, "MicroRNAs: genomics, biogenesis, mechanism, and function", *Cell*, vol. 116, pp. 281-297, 2004.

[3] N. Chomsky, "On certain formal properties of grammars", *Information and Control*, vol. 2, pp. 137-167, 1959.

[4] A. Condon, B. Davy, B. Rastegari, S. Zhao, and F. Tarrant, "Classifying RNA Pseudoknotted Structures", *Theoretical Computer Science*, vol.320, pp. 35-50, 2004.

[5] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis*, Cambridge Univ. Press, Cambridge, UK, 1998.

[6] S. R. Eddy and R. Durbin, "RNA sequence analysis using covariance models", *Nucleic Acids Research*, vol. 22, 2079-2088, 1994.

[7] S. R. Eddy, "Non-coding RNA genes and the modern RNA world", *Nature Reviews Genetics*, vol. 2, pp. 919-929, 2001.

[8] S. R. Eddy, "Computational genomics of noncoding RNA genes", *Cell*, vol. 109, pp. 127-140, 2002.

[9] V. A. Erdmann, M. Z. Barciszewska, M. Szymanski, A. Hochberg, N. de Groot, and J. Barciszewski, "The non-coding RNAs as riboregulators",*Nucleic Acids Research*, vol. 29, pp. 189-193, 2001.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.

[11] J. Gorodkin, L. J. Heyer, and G. D. Stormo, "Finding the most significant common sequence an structure motifs in a set of RNA sequences", *Nucleic Acids Research*, vol. 25, pp. 3724-3732, 1997.

[12] S. Gottesman, "Stealth regulation: biological circuitswith small RNA switches", *Genes & Development*, vol. 16, pp. 2829-2842, 2002.

[13] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy, "Rfam: an RNA family database", *Nucleic Acids Research*, vol. 31, pp. 439-441, 2003.

[14] R. J. Klein and S. R. Eddy, "RSEARCH: finding homologs of single structured RNA sequences", *BMC Bioinformatics*, vol. 4, 44, 2003.

[15] H. Matsui, K. Sato, and Y. Sakakibara, "Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures", *Bioinformatics*, vol. 21, pp. 2611-2617, 2005.

[16] J. S. Mattick, "Challenging the dogma: the hidden layer of non-protein-coding RNAs in complex organisms", *BioEssays*, vol. 25, pp. 930-939, 2003.

[17] V. Moulton, "Tracking down noncoding RNAs", *Proc. Natl. Acad. Sci. USA*, vol. 102, pp. 2269-2270, 2005.

[18] E. Rivas and S. R. Eddy, "The language of RNA: a formal grammar that includes pseudoknots", *Bioinformatics*, vol. 16, pp. 334-340, 2000.

[19] J. Ruan, G. D. Stormo, and W. Zhang, "An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots", *Bioinformatics*, vol. 20, pp. 58-66, 2004.

[20] J. Ruan, G. D. Stormo, and W. Zhang, "ILM: a web server for predicting RNA secondary structures with pseudoknots", *Nucleic Acids Research*, vol. 32, W146-W149, 2004.

[21] Y. Sakakibara, "Pair hidden Markov models on tree structures", *Bioinformatics*, vol. 19, i232-i240, 2003.

[22] P. Y. Shi, M. A. Brinton, J. M. Veal, Y. Y. Zhong, and W. D. Wilson, "Evidence for the existence of a pseudoknot structure at the 3' terminus of the flavivirus genomic RNA", *Biochemistry*, vol. 35, pp. 4222-4230, 1996.

[23] L. Simpson, O. H. Thiemann, N. J. Savill, J. D. Alfonzo, and D. A. Maslov, "Evolution of RNA editing in trypanosome mitochondria", *Proc, Natl. Acad. Sci., USA*, vol. 97, pp. 6986-6993, 2000.

[24] F. Sleutels, R. Zwart, and D. P. Barlow, "The non-coding Air RNA is required for silencing autosomal imprinted genes ", *Nature*, vol. 415, pp. 810-813, 2002.

[25] G. Storz, "An expanding universe of noncoding RNAs", *Science*, vol. 296, pp. 1260-1263, 2002.

[26] B. J. Tucker and R. R. Breaker, "Riboswitches as versatile gene control elements", *Current Opinion in Structural Biology*, vol. 15, pp. 342-348, 2005.

[27] F. H. D. van Batenburg, A. P. Gultyaev, C. W. A. Pleij, J. Ng, and J. Oliehoek, "Pseudobase: a database with RNA pseudoknots", *Nucleic Acids Research*, vol. 28, pp. 201-204, 2000.

[28] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260-267, 1967.

[29] K. M. Wassarman, W. A. Zhang, and G. Storz, "Small RNAs in Escherichia coli", *Trends in Microbiology*, vol. 7, pp. 37-45, 1999.

[30] Z. Weinberg and W. L. Ruzzo, "Sequence-based heuristics for faster annotation of noncodingRNA families," *Bioinformatics*, vol. 22, pp. 35-39, 2006.

[31] C. L. Will and R. Lührmann, "Spliceosomal UsnRNP biogenesis, structure and function", *Curr. Opin. Cell Biol.*, vol. 13, 2001.

[32] B.-J. Yoon and P. P. Vaidyanathan, "HMM with auxiliary memory: A new tool for modeling RNA secondary structures", *Proc. 38th Asilomar Conference on Signals, Systems, and Computers*, Monterey, CA, Nov. 2004.

[33] B.-J. Yoon and P. P. Vaidyanathan, "RNA secondary structure prediction using context-sensitive hidden Markov models", *Proc. International Workshop on Biomedical Circuits and Systems (BioCAS)*, Singapore, Dec. 2004.

[34] B.-J. Yoon and P. P. Vaidyanthan, "An overview of the role of context-sensitive HMMs in the prediction of ncRNA genes", *Proc. IEEE Workshop on Statistical Signal Processing*, Bordeaux, France, July 2005.

[35] B.-J. Yoon and P. P. Vaidyanthan, "Optimal alignment algorithm for context-sensitive hidden Markov models", *Proc. 30th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, Mar. 2005.

[36] B.-J. Yoon and P. P. Vaidyanthan, "Scoring algorithm for context-sensitive HMMs with application to RNA secondary structure analysis", *IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS)*, Newport, RI, May 2005.

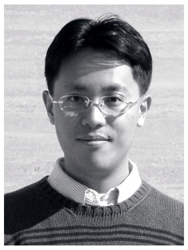
[37] B.-J. Yoon and P. P. Vaidyanathan, "Context-sensitive hidden Markov models for modeling long-range dependencies in symbol sequences", *IEEE Transactions on Signal Processing*, vol. 54, pp. 4169-4184, Nov. 2006.

[38] B.-J. Yoon and P. P. Vaidyanathan, "Profile context-sensitive HMMs for probabilistic modeling of sequences with complex correlations", *Proc. 31st International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, May 2006.

[39] B.-J. Yoon and P. P. Vaidyanthan, "Modeling and identification of alternative folding in regulatory RNAs using context-sensitive HMMs", *IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS)*, College Station, Texas, May 2006.

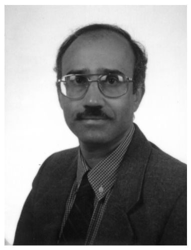
[40] B.-J. Yoon and P. P. Vaidyanathan, "Computational identification and analysis of noncoding RNAs - Unearthing the buried treasures in the genome", *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 64-74, Jan. 2007.

[41] B.-J. Yoon and P. P. Vaidyanathan, "Fast search of sequences with complex symbol correlations using profile context-sensitive HMMs and pre-screening filters", *Proc. 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, Apr. 2007.



Byung-Jun Yoon (S'02–M'07) was born in Seoul, Korea, in 1975. He received the B.S.E. (summa cum laude) degree from Seoul National University (SNU), Seoul, Korea in 1998 and the M.S. and Ph.D. degrees from California Institute of Technology (Caltech), Pasadena, CA, in 2002 and 2007, respectively, all in electrical engineering. He was a postdoctoral researcher at Caltech from Dec. 2006 to Oct. 2007. In 2008, he joined the Department of Electrical and Computer Engineering at the Texas A&M University, College Station, TX, where he is currently an assistant professor.

His main research interest is in bioinformatics, genomic signal processing, and systems biology. He received the Killgore Fellowship in 2001 from Caltech, and he was selected as a recipient of the Microsoft Research Graduate Fellowship for the year of 2004–2005. In 2003, he was awarded a prize in the student paper contest in the 37th Asilomar conference on signals, systems, and computers.



P. P. Vaidyanathan (S'80–M'83–SM'88–F'91) received the B.Sc. (Hons.) degree in Physics and the B.Tech. and M.Tech. degrees in Radio Physics and Electronics, all from the University of Calcutta, India, in 1974, 1977 and 1979, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from the University of California at Santa Barbara in 1982 and served as a postdoctoral fellow at the University of California, Santa Barbara from September 1982 to March 1983. Dr. Vaidyanathan joined the Electrical Engineering Department at the California Institute of Technology in 1983, and has been a full professor since 1993. Dr. Vaidyanathan has served as Executive Officer for the Electrical Engineering Department at Caltech from 2002-2005. His main research interests are in digital signal processing, multirate systems, signal processing for digital communications, system theory, and genomic signal processing.

Prof. Vaidyanathan served as Vice-Chairman of the Technical Program committee for the 1983 IEEE International Symposium on Circuits and Systems, and as the Technical Program Chairman for the 1992 IEEE International symposium on Circuits and Systems. He was an Associate Editor for the IEEE Transactions on Circuits and Systems for the period 1985-1987, and later as an Associate Editor for the journal IEEE Signal Processing letters, and a consulting editor for the journal *Applied and Computational Harmonic Analysis*. In 1992, he was a guest editor for special issues of the IEEE Trans. on Signal Processing and the IEEE Trans. on Circuits and Systems II, on the topics of filter banks, wavelets and sub-band coders.

Prof. Vaidyanathan has authored more than 360 papers in journals and conferences, and is the author of the book entitled *Multirate Systems and Filter Banks* (Englewood Cliffs, NJ: Prentice Hall 1993). He has written several chapters for various signal-processing handbooks. He is a recipient of the Award for Excellence in Teaching at the California Institute of Technology for the years 1983-1984, 1992-93 and 1993-94. In 1986, he received the NSF's Presidential Young Investigator award. In 1989, he received the IEEE ASSP Senior Award for his paper on multirate perfect-reconstruction filter banks. In 1990, he was recipient of the S. K. Mitra Memorial Award from the Institute of Electronics and Telecommunications Engineers, India, for his joint paper in the IETE journal. He was also the co-author of a paper on linear-phase perfect reconstruction filter banks in the IEEE SP Transactions, for which the first author (Truong Nguyen) received the Young outstanding author award in 1993. Dr. Vaidyanathan was elected IEEE Fellow in 1991.

In 1995, Dr. Vaidyanathan received the F. E. Terman Award of the American Society for Engineering Education, sponsored by Hewlett Packard Co., for his contributions to engineering education, especially his book *Multirate Systems and Filter Banks*. Dr. Vaidyanathan has given several plenary talks including the IEEE ISCAS-04, Sampta-01, Eusipco-98, SPCOM-95, and Asilomar-88 conferences on signal processing. He has been chosen a distinguished lecturer for the IEEE Signal Processing Society 1996-97. In 1999, he was chosen to receive the IEEE CAS Society's Golden Jubilee Medal. In 2002, he received the IEEE Signal Processing Society's Technical Achievement Award.