Hidden Markov Models and their Applications in Biological Sequence Analysis

Byung-Jun Yoon*

Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA

Abstract: Hidden Markov models (HMMs) have been extensively used in biological sequence analysis. In this paper, we give a tutorial review of HMMs and their applications in a variety of problems in molecular biology. We especially focus on three types of HMMs: the profile-HMMs, pair-HMMs, and context-sensitive HMMs. We show how these HMMs can be used to solve various sequence analysis problems, such as pairwise and multiple sequence alignments, gene annotation, classification, similarity search, and many others.

Received on: December 04, 2008 - Revised on: February 28, 2009 - Accepted on: March 02, 2009

Key Words: Hidden Markov model (HMM), pair-HMM, profile-HMM, context-sensitive HMM (csHMM), profile-csHMM, sequence analysis.

1. INTRODUCTION

The successful completion of many genome sequencing projects has left us with an enormous amount of sequence data. The sequenced genomes contain a wealth of invaluable information that can help us better understand the underlying mechanisms of various biological functions in cells. However, considering the huge size of the available data, it is virtually impossible to analyze them without the help of computational methods. In order to extract meaningful information from the data, we need computational techniques that can efficiently analyze the data according to sound mathematical principles. Given the expanding list of newly sequenced genomes and the increasing demand for genome re-sequencing in various comparative genomics projects, the importance of computational tools in biological sequence analysis is expected to grow only further.

Until now, various signal processing models and algorithms have been used in biological sequence analysis, among which the hidden Markov models (HMMs) have been especially popular. HMMs are well-known for their effectiveness in modeling the correlations between adjacent symbols, domains, or events, and they have been extensively used in various fields, especially in speech recognition [1] and digital communication. Considering the remarkable success of HMMs in engineering, it is no surprise that a wide range of problems in biological sequence analysis have also benefited from them. For example, HMMs and their variants have been used in gene prediction [2], pairwise and multiple sequence alignment [3, 4], base-calling [5], modeling DNA sequencing errors [6], protein secondary structure prediction [7], ncRNA identification [8], RNA structural alignment [9], acceleration of RNA folding and alignment [10], fast noncoding RNA annotation [11], and many others.

*Address correspondence to this author at the Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA; E-mail: bjyoon@ece.tamu.edu

In this paper, we give a tutorial review of HMMs and their applications in biological sequence analysis. The organization of the paper is as follows. In Sec. 2, we begin with a brief review of HMMs and the basic problems that must be addressed to use HMMs in practical applications. Algorithms for solving these problems are also introduced. After reviewing the basic concept of HMMs, we introduce three types of HMM variants, namely, profile-HMMs, pair-HMMs, and context-sensitive HMMs, that have been useful in various sequence analysis problems. Section 3 provides an overview of profile hidden Markov models and their applications. We also introduce publicly available profile-HMM software packages and libraries of pre-built profile-HMMs for known sequence families. In Sec. 4, we focus on pair-HMMs and their applications in pairwise alignment, multiple sequence alignment, and gene prediction. Section 5 reviews context-sensitive HMMs (csHMMs) and profile context-sensitive HMMs (profile-csHMMs), which are especially useful for representing RNA families. We show how these models and other types of HMMs can be employed in RNA sequence analysis.

2. HIDDEN MARKOV MODELS

A *hidden Markov model (HMM)* is a statistical model that can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable. We call the observed event a `symbol' and the invisible factor underlying the observation a `state'. An HMM consists of two stochastic processes, namely, an invisible process of hidden states and a visible process of observable symbols. The hidden states form a *Markov chain*, and the probability distribution of the observed symbol depends on the underlying state. For this reason, an HMM is also called a doubly-embedded stochastic process [1].

Modeling observations in these two layers, one visible and the other invisible, is very useful, since many real world problems deal with classifying raw observations into a number of categories, or class labels, that are more

1389-2029/09 $55.00+.00

meaningful to us. For example, let us consider the speech recognition problem, for which HMMs have been extensively used for several decades [1]. In speech recognition, we are interested in predicting the uttered word from a recorded speech signal. For this purpose, the speech recognizer tries to find the sequence of phonemes (states) that gave rise to the actual uttered sound (observations). Since there can be a large variation in the actual pronunciation, the original phonemes (and ultimately, the uttered word) cannot be directly observed, and need to be predicted.

This approach is also useful in modeling biological sequences, such as proteins and DNA sequences. Typically, a biological sequence consists of smaller substructures with different functions, and different functional regions often display distinct statistical properties. For example, it is well-known that proteins generally consist of multiple domains. Given a new protein, it would be interesting to predict the constituting domains (corresponding to one or more states in an HMM) and their locations in the amino acid sequence (observations). Furthermore, we may also want to find the protein family to which this new protein sequence belongs. In fact, HMMs have been shown to be very effective in representing biological sequences [3], as they have been successfully used for modeling speech signals. As a result, HMMs have become increasingly popular in computational molecular biology, and many state-of-the-art sequence analysis algorithms have been built on HMMs.

2.1. Definition

Let us now formally define an HMM. We denote the observed symbol sequence as $\mathbf{x} = x_1 x_2 \ldots x_L$ and the underlying state sequence as $\mathbf{y} = y_1 y_2 \ldots y_L$, where y_n is the underlying state of the n th observation x_n. Each symbol x_n takes on a finite number of possible values from the set of observations $\mathsf{O} = \{O_1, O_2, \ldots, O_N\}$, and each state y_n takes one of the values from the set of states $\mathsf{S} = \{1,2,\ldots,M\}$, where *N* and *M* denote the number of distinct observations and the number of distinct states in the model, respectively. We assume that the hidden state sequence is a time-homogeneous first-order Markov chain. This implies that the probability of entering state *j* in the next time point depends only on the current state *i*, and that this probability does not change over time. Therefore, we have

$$P\{y_{n+1} = j \mid y_n = i, y_{n-1} = i_{n-1}, \ldots, y_1 = i_1\} = P\{y_{n+1} = j \mid y_n = i\} = t(i,j) \quad (1)$$

for all states $i, j \in S$ and for all $n \geq 1$. The fixed probability for making a transition from state i to state j is called the *transition probability*, and we denote it by $t(i,j)$. For the initial state y_1, we denote the *initial state probability* as $\pi(i) = P\{y_1 = i\}$, for all $i \in S$. The probability that the n th observation will be $x_n = x$ depends only on the underlying state y_n, hence

$$P\{x_n = x \mid y_n = i, y_{n-1}, x_{n-1}, \ldots\} = P\{x_n = x \mid y_n = i\} = e(x \mid i) \quad (2)$$

for all possible observations $x \in O$, all state $i \in S$, and all $n \geq 1$. This is called the *emission probability* of x at state i, and we denote it by $e(x \mid i)$. The three probability measures $t(i,j)$, $\pi(i)$, and $e(x \mid i)$ completely specify an HMM. For convenience, we denote the set of these parameters as Θ.

Based on these parameters, we can now compute the probability that the HMM will generate the observation sequence $\mathbf{x} = x_1 x_2 \ldots x_L$ with the underlying state sequence $\mathbf{y} = y_1 y_2 \ldots y_L$. This joint probability $P\{\mathbf{x}, \mathbf{y} \mid \Theta\}$ can be computed by

$$P\{\mathbf{x}, \mathbf{y} \mid \Theta\} = P\{\mathbf{x} \mid \mathbf{y}, \Theta\} P\{\mathbf{y} \mid \Theta\}, \quad (3)$$

where

$$P\{\mathbf{x} \mid \mathbf{y}, \Theta\} = e(x_1 \mid y_1) e(x_2 \mid y_2) e(x_3 \mid y_3) \cdots e(x_L \mid y_L) \quad (4)$$

$$P\{\mathbf{y} \mid \Theta\} = \pi(y_1) t(y_1, y_2) t(y_2, y_3) \cdots t(y_{L-1}, y_L). \quad (5)$$

As we can see, computing the observation probability is straightforward when we know the underlying state sequence.

2.2. A Simple HMM for Modeling Eukaryotic Genes

As we mentioned earlier, HMMs can be effectively used for representing biological sequences. As a simple example, let us consider an HMM that models protein-coding genes in eukaryotes. It is well known that many protein-coding regions display codon bias. The nonuniform usage of codons results in different symbol statistics for different codon positions [12], and it is also a source of the period-3 property in the coding regions [13]. These properties are not observed in introns, which are not translated into amino acids. Therefore, it is important to incorporate these codon statistics when modeling protein-coding genes and building a gene-finder. Fig. (**1**) shows a toy HMM for modeling eukaryotic genes. The given HMM tries to capture the statistical differences in exons and introns. The HMM has four states, where E_1, E_2, and E_3 are used to model the base statistics in exons. Each E_k uses a different set of emission probabilities to reflect the symbol statistics at the k th position of a codon. The state I is used to model the base statistics in introns. Note that this HMM can represent genes with multiple exons, where the respective exons can have variable number of codons, and the introns can also have variable lengths. This example shows that if we know the structure and the important characteristics of the biological sequences of interest, building the corresponding HMM is relatively simple and it can be done in an intuitive manner.

The constructed HMM can now be used to analyze new observation sequences. For example, let us assume that we have a new DNA sequence $\mathbf{x} = x_1 \cdots x_{19} =$ ATGCGACTGCATAGCACTT How can we find out whether this DNA sequence is a coding gene or not? Or, if

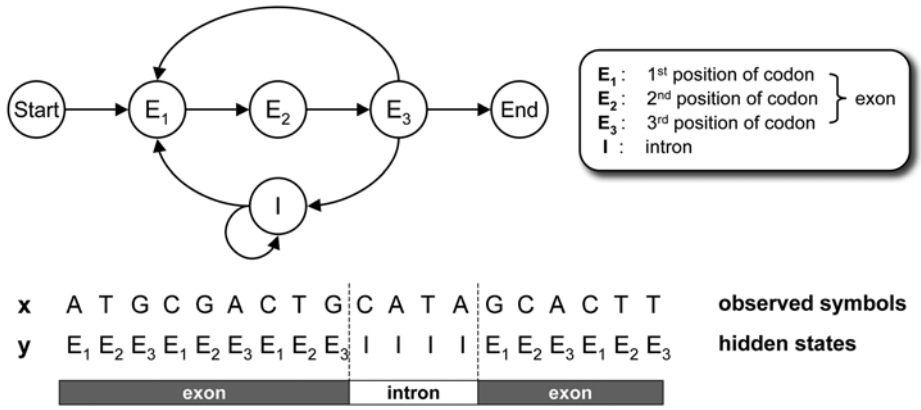

Fig. (1). A simple HMM for modeling eukaryotic genes.

we assume that $\mathbf{x}$ is a protein-coding gene, how can we predict the locations of the exons and introns in the given sequence? We can answer the first question by computing the observation probability of $\mathbf{x}$ based on the given HMM that models coding genes. If this probability is high, it implies that this DNA sequence is likely to be a coding gene. Otherwise, we may conclude that $\mathbf{x}$ is unlikely to be a coding gene, since it does not contain the statistical properties that are typically observed in protein-coding genes. The second question is about predicting the internal structure of the sequence, as it cannot be directly observed. To answer this question, we may first predict the state sequence $\mathbf{y}$ in the HMM that best describes $\mathbf{x}$. Once we have inferred the best $\mathbf{y}$, it is straightforward to predict the locations of the exons and introns. For example, assume that the optimal state sequence $\mathbf{y}$ is as shown in Fig. (**1**). This implies that the first nine bases $x_1 \cdots x_9$ belong to the first exon, the following four bases $x_{10} \cdots x_{13}$ belong to an intron, and the last six bases $x_{14} \cdots x_{19}$ belong to another exon. As these examples show, HMMs provide a formal probabilistic framework for analyzing biological sequences.

2.3. Basic Problems and Algorithms for HMMs

There are three basic problems that have to be addressed in order to use HMMs in practical applications. Suppose we have a new symbol sequence $\mathbf{x} = x_1 x_2 \ldots x_L$. How can we compute the observation probability $P\{\mathbf{x} \mid \Theta\}$ based on a given HMM? This problem is sometimes called the *scoring problem*, since computing the probability $P\{\mathbf{x} \mid \Theta\}$ is a natural way of `scoring' a new observation sequence $\mathbf{x}$ based on the model at hand. Note that for a given $\mathbf{x}$, its underlying state sequence is not directly observable and there can be many state sequences that yield $\mathbf{x}$. Therefore, one way to compute the observation probability is to consider all possible state sequences $\mathbf{y}$ for the given $\mathbf{x}$ and sum up the probabilities as follows

$$P\{\mathbf{x} \mid \Theta\} = \sum_{\mathbf{y}} P\{\mathbf{x}, \mathbf{y} \mid \Theta\}. \tag{6}$$

However, this is computationally very expensive, since there are M^L possible state sequences. For this reason, we definitely need a more efficient method for computing $P\{\mathbf{x} \mid \Theta\}$. There exist a dynamic programming algorithm, called the *forward algorithm*, that can compute $P\{\mathbf{x} \mid \Theta\}$ in an efficient manner [1]. Instead of enumerating all possible state sequences, this algorithm defines the following *forward variable*

$$\alpha(n,i) = P\{x_1 \cdots x_n, y_n = i \mid \Theta\}. \tag{7}$$

This variable can be recursively computed using the following formula

$$\alpha(n,i) = \sum_k [\alpha(n-1,k) t(k,i) e(x_n \mid i)], \tag{8}$$

for $n = 2, \ldots, L$. At the end of the recursions, we can compute $P\{\mathbf{x} \mid \Theta\} = \sum_k \alpha(L,k)$. This algorithm computes the observation probability of $\mathbf{x}$ with only $O(LM^2)$ computations. Therefore, the amount of time required for computing the probability increases only linearly with the sequence length L, instead of increasing exponentially.

Another practically important problem is to find the optimal state sequence, or the optimal path, in the HMM that maximizes the observation probability of the given symbol sequence $\mathbf{x}$. Among all possible state sequences $\mathbf{y}$, we want to find the state sequence that best explains the observed symbol sequence. This can be viewed as finding the best alignment between the symbol sequence and the HMM, hence it is sometimes called the *optimal alignment* problem. Formally, we want to find the optimal path $\mathbf{y}^*$ that satisfies the following

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} P\{\mathbf{y} \mid \mathbf{x}, \Theta\}. \tag{9}$$

Note that this is identical to finding the state sequence that maximizes $P\{\mathbf{x}, \mathbf{y} \mid \Theta\}$, since we have

$$P\{\mathbf{y} \mid \mathbf{x}, \Theta\} = \frac{P\{\mathbf{x}, \mathbf{y} \mid \Theta\}}{P\{\mathbf{x} \mid \Theta\}}. \tag{10}$$

Finding the optimal state sequence $\mathbf{y}^*$ by comparing all M^L possible state sequences is computationally infeasible. However, we can use another dynamic programming algorithm, well-known as the *Viterbi algorithm*, to find the optimal path $\mathbf{y}^*$ efficiently [14, 15]. The Viterbi algorithm defines the variable

$$\gamma(n,i) = \max_{y_1,\ldots,y_{n-1}} P\{x_1 \cdots x_n, y_1 \cdots y_{n-1} y_n = i \mid \Theta\}, \quad (11)$$

and computes it recursively using the following formula

$$\gamma(n,i) = \max_k [\gamma(n-1,k) t(k,i) e(x_n \mid i)]. \quad (12)$$

At the end, we can obtain the maximum observation probability as follows

$$P^* = \max_{\mathbf{y}} P\{\mathbf{x},\mathbf{y} \mid \Theta\} = \max_k \gamma(L,k). \quad (13)$$

The optimal path $\mathbf{y}^*$ can be easily found by tracing back the recursions that led to the maximum probability $P^* = P\{\mathbf{x},\mathbf{y}^* \mid \Theta\}$. Like the forward algorithm, the Viterbi algorithm finds the optimal state sequence in $O(LM^2)$ time.

As we have seen, the Viterbi algorithm finds the optimal path that maximizes the observation probability of the *entire* symbol sequence. In some cases, it may be more useful to find the optimal states individually for each symbol position. In this case, we can find the optimal state y_n that is most likely to be the underlying state of x_n as follows

$$\hat{y}_n = \arg\max_i P\{y_n = i \mid \mathbf{x}, \Theta\}, \quad (14)$$

based on the given $\mathbf{x}$ and Θ. The posterior probability $P\{y_n = i \mid \mathbf{x}, \Theta\}$ can be computed from

$$P\{y_n = i \mid \mathbf{x}, \Theta\} = \frac{P\{x_1 \cdots x_n, y_n = i \mid \Theta\} P\{x_{n+1} \cdots x_L \mid y_n = i, \Theta\}}{P\{\mathbf{x} \mid \Theta\}}$$

$$= \frac{\alpha(n,i)\beta(n,i)}{\sum_k \alpha(n,k)\beta(n,k)}, \quad (15)$$

where $\beta(n,i)$ is defined as

$$\beta(n,i) = P\{x_{n+1} \cdots x_L \mid y_n = i, \Theta\}. \quad (16)$$

This *backward variable* $\beta(n,i)$ can be recursively computed using the *backward algorithm* as follows

$$\beta(n,i) = \sum_k [t(i,k) e(x_{n+1} \mid k) \beta(n+1,k)], \quad (17)$$

for $n = L-1, L-2, \ldots, 1$. The advantage of predicting the optimal states individually is that this approach will maximize the expected number of correctly predicted states. However, the overall state sequence $\hat{\mathbf{y}} = \hat{y}_1 \hat{y}_2 \cdots \hat{y}_L$ will be generally suboptimal, hence $P\{\mathbf{x},\hat{\mathbf{y}} \mid \Theta\} \le P\{\mathbf{x},\mathbf{y}^* \mid \Theta\}$. In some cases, the predicted path $\hat{\mathbf{y}}$ may not be even a legitimate path in the given HMM, in which case we will have $P\{\mathbf{x},\hat{\mathbf{y}} \mid \Theta\} = 0$. For this reason, the Viterbi algorithm is often preferred when we are interested in inferring the optimal state sequence for the entire observation $\mathbf{x}$, while the posterior-decoding approach in (14) is preferred when our interest is mainly in predicting the optimal state at a specific position. The posterior probability in (15) can also be useful for estimating the reliability of a state prediction. For example, we may first predict the optimal path $\mathbf{y}^* = y_1^* \cdots y_L^*$ as in (9) using the Viterbi algorithm, and then estimate the reliability of the individual state prediction y_n^* by computing the posterior probability $P\{y_n = y_n^* \mid \mathbf{x}, \Theta\}$ as in (15).

The scoring problem and the alignment problem are concerned about analyzing a new observation sequence $\mathbf{x}$ based on the given HMM. However, the solutions to these problems are meaningful only if the HMM can properly represent the sequences of our interest. Let us assume that we have a set of related observation sequences $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ that we want to represent by an HMM. For example, they may be different speech recordings of the same word or protein sequences that belong to the same functional family. Now, the important question is how we can reasonably choose the HMM parameters based on these observations. This is typically called the *training problem*. Although there is no optimal way of estimating the parameters from a limited number of finite observation sequences, there are ways to find the HMM parameters that locally maximize the observation probability [1, 16-18]. For example, we can use the *Baum-Welch* algorithm [16] to train the HMM. The Baum-Welch algorithm is an expectation-maximization (EM) algorithm that iteratively estimates and updates Θ based on the *forward-backward* procedure [1, 16]. Since the estimation of the HMM parameters is essentially an optimization problem, we can also use standard gradient-based techniques to find the optimal parameters of the HMM [17, 18]. It has been demonstrated that the gradient-based method can yield good estimation results that are comparable to those of the popular EM-based method [18]. When the precise evaluation of the probability (or likelihood) of an observation is practically intractable for the HMM at hand, we may use simulation-based techniques to evaluate it approximately [17, 19]. These techniques allow us to handle a much broader class of HMMs. In such cases, we can train the HMM using the *Monte Carlo EM (MCEM)* algorithm, which adopts the Monte Carlo approach to approximate the so-called E-step (expectation step) in the EM algorithm [19]. There are also training methods based on stochastic optimization algorithms, such as simulated annealing, that try to improve the optimization results by avoiding local maxima [20, 21]. Currently, there exists a vast literature on estimating the parameters of hidden Markov models, and the reader is referred to [1, 17, 19, 22, 23] for further discussions.

2.4. Variants of HMMs

There exist a large number of HMM variants that modify and extend the basic model to meet the needs of various applications. For example, we can add silent states (i.e., states that do not emit any symbol) to the model in order to represent the absence of certain symbols that are expected to be present at specific locations [24, 25]. We can also make the states emit two aligned symbols, instead of a single symbol, so that the resulting HMM simultaneously generates two related symbol sequences [3, 4, 26]. It is also possible to make the probabilities at certain states dependent on part of the previous emissions [9, 27] so that we can describe more complex symbol correlations. In the following sections, we review a number of HMM variants that have been used in various biological sequence analysis problems.

3. PROFILE HIDDEN MARKOV MODELS

Let us assume that we have a multiple sequence alignment of proteins or DNA sequences that belong to the same functional family. How can we build an HMM that can effectively represent the common patterns, motifs, and other statistical properties in the given alignment? One model that is especially useful for representing the profile of a multiple sequence alignment is the *profile hidden Markov model (profile-HMM)* [24, 25]. Profile-HMMs are HMMs with a specific architecture that is suitable for modeling sequence profiles. Unlike general HMMs, profile-HMMs have a strictly linear left-to-right structure that does not contain any cycles. A profile-HMM repetitively uses three types of hidden states, namely, *match states* M_k, *insert states* I_k, and *delete states* D_k, to describe position-specific symbol frequencies, symbol insertions, and symbol deletions, respectively.

3.1. Constructing a Profile-HMM

To see how profile-HMMs work, let us consider the following example. Suppose we want to construct a profile-HMM based on the multiple alignment shown in Fig. (**2a**).

As we can see, the given alignment has five columns, where the base frequencies in the respective columns are different from each other. The k th match state M_k in the profile-HMM is used to describe the symbol frequencies in the k th column of the alignment. It is called a `match' state, since it is used to represent the case when a symbol in a new observation sequence matches the k th symbol in the consensus sequence of the original alignment. As a result, the number of match states in the resulting profile-HMM is identical to the length of the consensus sequence. The emission probability $e(x \mid M_k)$ at the k th match state M_k reflects the observed symbol frequencies in the k th consensus column. By interconnecting the match states $M_1, M_2, \ldots, M_5$, we obtain an *ungapped HMM* as shown in Fig. (**2b**). This ungapped HMM can represent DNA sequences that match the consensus sequence of the alignment without any gap, and it serves as the backbone of the final profile-HMM that is to be constructed.

Once we have constructed the ungapped HMM, we add insert states I_k and delete states D_k to the model so that we can account for insertions and deletions in new observation sequences. Let us first consider the case when the observed DNA sequence is longer than the consensus sequence of the original alignment. In this case, if we align these sequences, there will be one or more bases in the observed DNA sequence that are not present in the consensus sequence. These additional symbols are modeled by the insert states. The insert state I_k is used to handle the symbols that are inserted between the k th and the $(k+1)$ th positions in the consensus sequence. Now, let us consider the case when the new observed sequence is shorter than the consensus sequence. In this case, there will be one or more bases in the consensus sequence that are not present in the observed DNA sequence. The k th delete state D_k is used to handle the deletion of the k th symbol in the original consensus sequence. As delete states represent symbols that are missing, D_k is a *non-emitting state*, or a *silent* state, which is simply used as a place-holder that interconnects the neighboring states. After adding the insert states and the delete states to the ungapped HMM in Fig. (**2b**), we obtain the final profile-HMM that is shown in Fig. (**2c**).

Estimating the parameters of a profile-HMM based on a given multiple sequence alignment is relatively simple. We first have to decide which columns should be represented by match states and which columns should be modeled by insert states. Suppose we have a column that contains one or more gaps. Should we regard the symbols in the column as `insertions', or should we rather view the gaps in the column as `deletions'? One simple rule would be to compare the number of symbols and the number of gaps. If the column has more symbols than gaps, we treat the gaps as symbol deletions. Therefore, we model the column using a match state M_k (for the symbols in the given column) and a delete state D_k (for the gaps in the same column). On the contrary, if we have more gaps than symbols, it would make more sense to view the symbols as insertions, hence we use an insert state I_k to represent the column. Once we have decided which columns should be represented by match states and which ones should be represented by insert states, we know the underlying state sequence for each symbol sequence in the alignment. Therefore, we can estimate the transition probabilities and the emission probabilities of the profile-HMM by counting the number of each state transition or symbol emission and computing their relative frequencies. To allow small probability for state transitions or symbol emissions that are not observed in the original alignment, we can add the so-called *pseudocounts* to the actual counts [3].

We can also use more sophisticated methods for parameterizing the profile-HMMs. In fact, there have been considerable research efforts for optimal construction and parameterization of profile-HMMs to improve their overall performance. More discussions on this topic can be found in [3, 28-32].

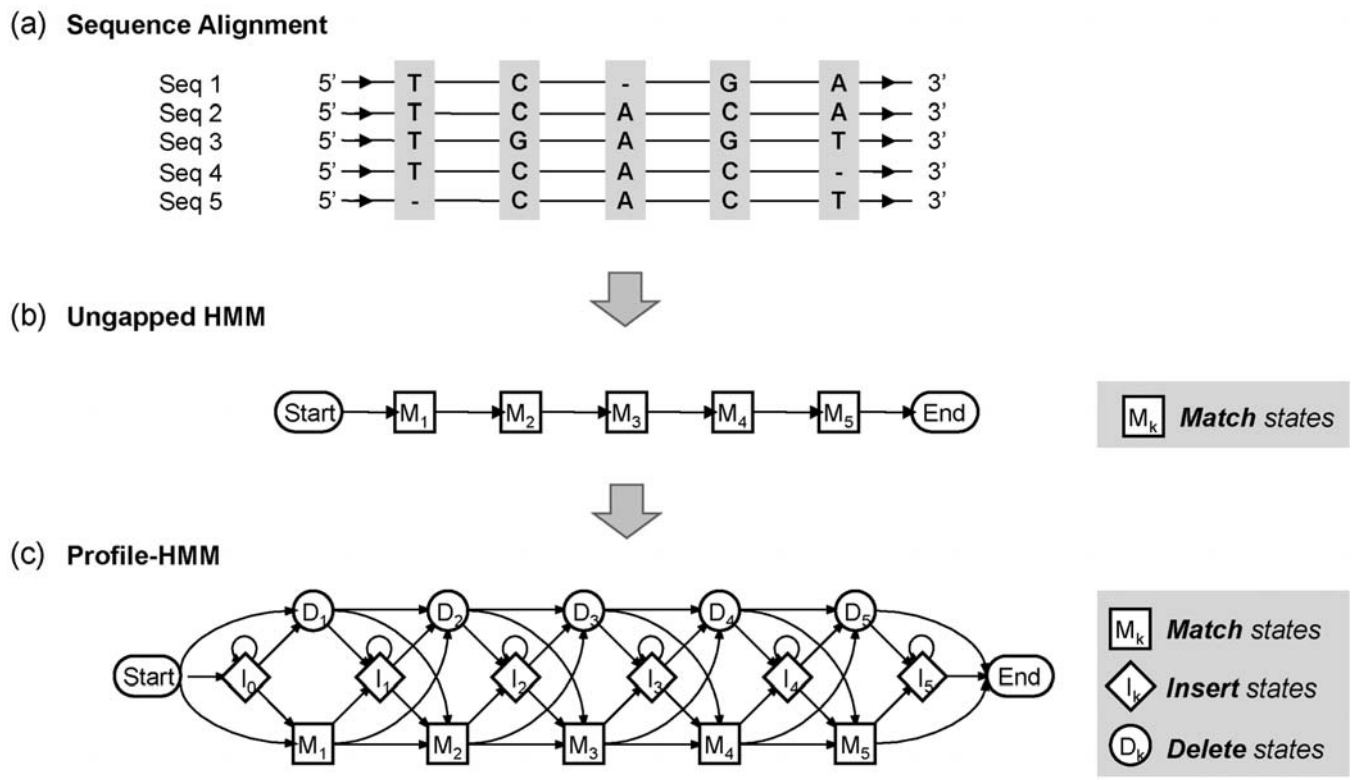

Fig. (2). Profile hidden Markov model. (**a**) Multiple sequence alignment for constructing the profile-HMM. (**b**) The ungapped HMM that represents the consensus sequence of the alignment. (**c**) The final profile-HMM that allows insertions and deletions.

3.2. Applications of Profile-HMMs

Due to the convenience and effectiveness in representing sequence profiles, profile-HMMs have been widely used for modeling and analyzing biological sequences. When profile-HMMs were first proposed, they were quickly adopted for modeling the characteristics of a number of protein families, such as globins, immunoglobulins, and kinases [33]. They have been shown to be useful for various tasks, including protein classification, motif detection, and finding multiple sequence alignments. Nowadays, there exist publicly available software packages, such as HMMER [3] and SAM [34, 35], that can be readily used to build and train profile-HMMs. These packages provide convenient tools for applying profile-HMMs to various sequence analysis problem. A comparison between these two popular HMM packages and an assessment of their critical features can be found in [32].

It would be also very convenient to have a library of ready-made profile-HMMs for known sequence families. Currently, we have two such libraries that have compiled a large number of profile-HMMs for various protein families: the PROSITE database [36, 37] and the Pfam database [38, 39]. Given a profile-HMM that represents a biological sequence family, we can use it to search a sequence database to find additional homologues that belong to the same family. In a similar manner, if we have a database of pre-built profile-HMMs, we can use a single query sequence to search through the database to look for matching profiles. This strategy can be used for classification and annotation of the given sequence. For example, by querying a new protein sequence against Pfam or PROSITE, we can find out whether the sequence contains any of the known protein domains.

Sometimes, we may want to compare two multiple sequence alignments or sequence profiles, instead of comparing a single sequence against a multiple alignment or a profile. Comparing sequence profiles can be beneficial for detecting remote homologues, and profile-HMMs have also been used for this purpose [40-42]. For example, COACH [40] allows us to compare sequence alignments, by building a profile-HMM from one alignment and aligning the other alignment to the constructed profile-HMM. HHsearch [42] generalizes the traditional pairwise sequence alignment algorithm for finding the alignment of two profile-HMMs. Another program, called PRC (profile comparer) [41], provides a tool for scoring and aligning profile-HMMs produced by popular software tools, including HMMER [3] and SAM [34, 35].

Although profile-HMMs have been widely used for representing sequence profiles, their application is by no means limited to modeling amino acid or nucleotide sequences. For example, Di Francesco *et al*. [43, 44] used profile-HMMs to model sequences of protein secondary structure symbols: helix (H), strand (E), and coil (C). Therefore, the model emits only three types of symbols instead of twenty different amino acids. It has been demonstrated that this profile-HMM can be used for recognizing the three-dimensional fold of new protein sequences based on their secondary structure predictions. Another interesting example is the *feature-based profile-*

HMM that was proposed to improve the performance of remote protein homology detection [45]. Instead of emitting amino acids, emissions of these HMMs are based on `features' that capture the biochemical properties of the protein family of interest. These features are extracted by performing a spectral analysis of a number of selected `amino acid indices' [46] and using principal component analysis (PCA) to reduce the redundancy in the resulting signal.

There are also variants of the basic profile-HMM, where the *jumping profile-HMM (jpHMM)* [47] is one such example. The jumping profile-HMM is a probabilistic generalization of the so-called *jumping-alignment* approach. The jumping-alignment approach is a strategy for comparing a sequence with a multiple alignment, where the sequence is not aligned to the alignment as a whole, but it can `jump' between the sequences that constitute the alignment. In this way, different parts of the sequence can be aligned to different sequences in the given alignment. A jpHMM uses multiple match states for each column to represent different sequence subtypes. The HMM is allowed to jump between these match states based on the local similarity of the sequence and the different sequence subtypes in the model. This approach has been shown to be especially useful for detecting recombination breakpoints [47].

4. PAIR HIDDEN MARKOV MODELS

In biological sequence analysis, it is often important to compare two sequences to find out whether these sequences are functionally related. Sequence similarity is often a good indicator of their functional relevance, and for this reason, methods for quantitatively measuring the similarity of two proteins or DNA sequences have been of interest to many researchers. A typical approach for comparing two biological sequences is to align them based on their similarity, compute their alignment score, and evaluate the statistical significance of the predicted alignment. To find the best alignment between the sequences, we first have to define a reasonable scoring scheme for ranking different alignments. Based on this scoring scheme, we can choose the alignment that maximizes the alignment score.

4.1. Pair-HMMs for Modeling Aligned Sequence Pairs

The *pair hidden Markov model (pair-HMM)* [3] is a variant of the basic HMM that is especially useful for finding sequence alignments and evaluating the significance of the aligned symbols. Unlike the original HMM, which generates only a single sequence, a pair-HMM generates an aligned pair of sequences. For example, let us consider the pair-HMM shown in Fig. (**3**).

This simple pair-HMM traverses between the states I_X, I_Z, and A, to simultaneously generate two aligned DNA sequences $\mathbf{x} = x_1 \cdots x_{L_x}$ (sequence 1) and $\mathbf{z} = z_1 \cdots z_{L_z}$ (sequence 2). The state I_X emits a single unaligned symbol x_i in the first sequence $\mathbf{x}$. Similarly, the state I_Z emits an unaligned symbol z_j only in the second sequence $\mathbf{z}$. Finally, the state A generates an aligned pair of two symbols x_i and z_j, where x_i is inserted in $\mathbf{x}$ and z_j is inserted in $\mathbf{z}$. For example, let us consider the alignment between $\mathbf{x} = x_1x_2x_3x_4x_5 = \text{TTCCG}$ and $\mathbf{z} = z_1z_2z_3z_4z_5 = \text{CCGTT}$ illustrated in Fig. (**3**). We assume that the underlying state sequence is $\mathbf{y} = \text{I}_\text{X}\text{I}_\text{X}\text{AAAI}_\text{Z}\text{I}_\text{Z}$ as shown in the figure. As we can see, x_1 and x_2 are individually emitted at I_X, hence they are not aligned to any bases in $\mathbf{z}$. The pairs (x_3, z_1), (x_4, z_2), and (x_5, z_3) are jointly emitted at A, and therefore the bases in the respective pairs are aligned to each other. Finally, z_4 and z_5 are individually emitted at I_z as unaligned bases.

As we can see from this example, there is a one-to-one relationship between the hidden state sequence $\mathbf{y}$ and the alignment between the two observed sequences $\mathbf{x}$ and $\mathbf{z}$. Therefore, based on the pair-HMM framework, the problem of finding the best alignment between $\mathbf{x}$ and $\mathbf{z}$ reduces to the problem of finding the following optimal state sequence

$$\mathbf{y}^* = \underset{\mathbf{y}}{arg\,max}\, P\{\mathbf{y} \mid \mathbf{x}, \mathbf{z}, \Theta\}. \quad (18)$$

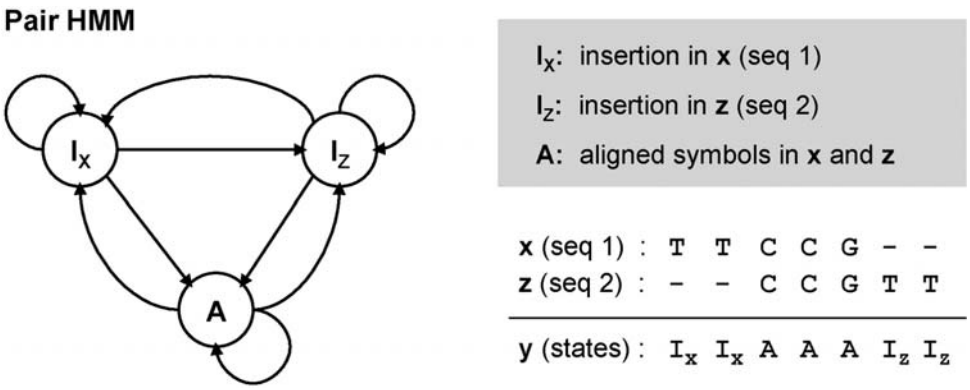

Fig. (3). Example of a pair hidden Markov model. A pair-HMM generates an aligned pair of sequences. In this example, two DNA sequences $\mathbf{x}$ and $\mathbf{z}$ are simultaneously generated by the pair-HMM, where the underlying state sequence is $\mathbf{y}$. Note that the state sequence $\mathbf{y}$ uniquely determines the pairwise alignment between $\mathbf{x}$ and $\mathbf{z}$.

Note that this is identical to finding the optimal path that maximizes $P\{\mathbf{x},\mathbf{z},\mathbf{y} \mid \Theta\}$, since we have

$$P\{\mathbf{y} \mid \mathbf{x},\mathbf{z},\Theta\} = \frac{P\{\mathbf{x},\mathbf{z},\mathbf{y} \mid \Theta\}}{P\{\mathbf{x},\mathbf{z} \mid \Theta\}}. \quad (19)$$

The optimal state sequence $\mathbf{y}^*$ can be found using dynamic programming, by a simple modification of the Viterbi algorithm [3]. The computational complexity of the resulting alignment algorithm is only $O(L_x L_z)$, where L_x and L_z are the lengths of $\mathbf{x}$ and $\mathbf{z}$, respectively.

An important advantage of the pair-HMM based approach over traditional alignment algorithms is that we can use the pair-HMM to compute the alignment probability of a sequence pair. When the given sequences do not display strong similarities, it is difficult to find the correct alignment that is biologically meaningful. In such cases, it would be more useful to compute the probability that the sequences are related, instead of focusing only on their best alignment. The joint observation probability $P\{\mathbf{x},\mathbf{z} \mid \Theta\}$ of sequences $\mathbf{x}$ and $\mathbf{z}$ can be computed by summing over all possible state sequences

$$P\{\mathbf{x},\mathbf{z} \mid \Theta\} = \sum_{\mathbf{y}} P\{\mathbf{x},\mathbf{z},\mathbf{y} \mid \Theta\}. \quad (20)$$

Instead of enumerating all possible state sequences, we can modify the original forward algorithm to compute $P\{\mathbf{x},\mathbf{z} \mid \Theta\}$ in an efficient manner [3]. It is also possible to compute the alignment probability for individual symbol pairs. For example, the probability that x_i will be aligned to z_j is $P(y_k = A \mid \mathbf{x},\mathbf{z},\Theta)$, where y_k denotes the underlying state for the aligned pair (x_i, z_j). This probability can be computed as follows

$$P\{y_k = A \mid \mathbf{x},\mathbf{z},\Theta\} = \frac{P\{x_1 \cdots x_i, z_1 \cdots z_j, y_k = A \mid \Theta\} P\{x_{i+1} \cdots x_{L_x}, z_{j+1} \cdots z_{L_z} \mid y_k = A, \Theta\}}{P\{\mathbf{x},\mathbf{z} \mid \Theta\}} \quad (21)$$

using a modified forward-backward algorithm [3].

4.2. Applications of Pair-HMMs

As pair-HMMs provide a full probabilistic framework for handling pairwise alignments, they have been extensively used for finding pairwise alignment of proteins and DNA sequences [3]. For example, the pair-HMM was used to approximate an explicit model for symbol insertions and deletions (indels) in [48]. The constructed pair-HMM was then used to find the optimal sequence alignment, compute the overall alignment probability, and estimate the reliability of the individual alignment regions. It was demonstrated that using geometrically distributed indel lengths based on pair-HMMs has many potential advantages [48]. More recently, another method called MCALIGN2 [49] also adopted pair-HMMs with a slightly different structure, for global pairwise alignment of noncoding DNA segements. Using pair-HMMs to describe specific indel length distributions has been shown to be very useful for finding accurate alignments of non-coding DNA sequences.

Many multiple sequence alignment (MSA) algorithms also make use of pair-HMMs [50-52]. The most widely adopted strategy for constructing a multiple alignment is the *progressive alignment* approach, where sequences are assembled into one large multiple alignment through consecutive pairwise alignment steps according to a *guide tree* [53, 54]. The algorithms proposed in [50-52] combine pair-HMMs with the progressive alignment approach to construct multiple sequence alignments. For example, the MSA algorithm in [51] uses a pair-HMM to find pairwise alignments and to estimate their alignment reliability. In addition to predicting the best multiple alignment, this method computes the minimum posterior probability for each column, which has been shown to correlate well with the correctness of the prediction. These posterior probabilities can be used to filter out the columns that are unreliably aligned. Another state-of-the-art MSA algorithm called ProbCons [50] also uses a pair-HMM to compute the posterior alignment probabilities. Instead of directly using the optimal alignment predicted by the Viterbi algorithm, ProbCons tries to find the pairwise alignment that maximizes the expected number of correctly aligned pairs based on the posterior probabilities. Furthermore, the algorithm incorporates multiple sequence conservation information when finding the pairwise alignments. This is achieved by using the match quality scores that are obtained from *probabilistic consistency transformation* of the posterior probabilities, when finding the alignments. It was demonstrated that this probabilistic consistency based approach can achieve significant improvement over traditional progressive alignment algorithms [50].

Pair-HMMs have also been used for gene prediction [4, 55-58]. For example, a method called Pairagon+N-SCAN_EST provides a convenient pipeline for gene annotation by combining a pair-HMM with a *de novo* gene prediction algorithm [56]. In this method, a pair-HMM is first used to find accurate alignments of cDNA sequences to a given genome, and these alignments are combined with a gene prediction algorithm for accurate genome annotation. A number of gene-finders adopt a comparative approach for gene prediction [4, 55, 57, 58]. The *generalized pair hidden Markov model (GPHMM)* [4] provides a convenient probabilistic framework for comparative gene prediction by combining the pair-HMM (widely used for sequence alignment and comparison) and the generalized HMM (used by many gene finders). Comparative gene-finders such as SLAM [55] and TWAIN [57] are implemented based on the GPHMM framework. A similar model has been also proposed in [58] to compare two DNA sequences and jointly analyze their gene structures.

Although the pair-HMM is originally defined on the pairwise alignment of *linear* symbol sequences, we can use it for aligning more complex structures, such as trees. For example, the PHMMTSs (pair hidden Markov models on tree structures) extend the pair-HMMs so that we can use

them for aligning trees [59]. As most RNA secondary structures can be represented by trees, PHMMTSs provide a useful probabilistic framework for aligning RNA sequences. In [59], PHMMTSs have been used to find the *structural alignment* of RNAs, where an RNA with an unknown structure is aligned to an RNA with a known secondary structure. This structural alignment is distinct from a sequence-based alignment, in the sense that we consider both the structural similarity and the sequence similarity when finding the optimal alignment between the RNAs. *Pair stochastic tree adjoining grammars (PSTAGs)* extend the PHMMTSs further, so that we can use them to align TAG (tree adjoining grammar) trees [60]. This extension allows us to align RNAs with more complicated secondary structures, including pseudoknots.

5. CONTEXT-SENSITIVE HMMS AND PROFILE-CSHMMS

Despite their usefulness in various sequence analysis problems, especially, those dealing with proteins and DNA sequences, traditional HMMs have inherent limitations that make them not suitable for handling RNA sequences. Many non-coding RNAs (ncRNAs) conserve base-paired secondary structures that induce pairwise correlations between non-adjacent bases [61]. However, traditional HMMs assume that the emission probability of each symbol depends solely on the underlying state, and since each state depends only on its previous state, they cannot effectively describe correlations between distant symbols. For this reason, more complex models such as the *stochastic context-free grammars (SCFGs)* have been employed in RNA sequence analysis [62, 63]. Although HMMs cannot be directly used for modeling RNAs, we can extend the original model to handle pairwise base correlations. The *context-sensitive HMM (csHMM)* is a variant of HMM that can be used for this purpose [27, 64].

5.1. Context-Sensitive Hidden Markov Models

The main difference between a context-sensitive HMM and a traditional HMM is that a csHMM can use part of the past emissions (called the `context') to adjust the probabilities at certain future states. The use of such contextual information is very useful in describing long-range correlations between symbols, and this context-dependency increases the descriptive capability of the HMM considerably [27]. Unlike traditional HMMs, csHMMs use three different types of hidden states: *single-emission states* S_n, *pairwise-emission states* P_n, and *context-sensitive states* C_n. The single-emission states are similar to the regular states in traditional HMMs. They have fixed emission probabilities and do not make use of any contextual information. In addition to the single-emission states, two new types of states, the pairwise-emission states and the context-sensitive states, are introduced in csHMMs. These states cooperate to describe pairwise symbol correlations. Like single-emission states, pairwise-emission states also have fixed emission probabilities. However, the symbols emitted at a pairwise-emission state P_n are stored in the memory[1] that is associated with the state P_n. These symbols are used later on as the `contextual information' for adjusting the probabilities at the corresponding context-sensitive state C_n. When we enter the context-sensitive state C_n, we first access the associated memory to retrieve the symbol x_i that was previously emitted at the corresponding pairwise-emission state $y_i = P_n$. The emission probability at $y_j = C_n$ ($j > i$) is adjusted based on the retrieved symbol x_i (the `context'). We can denote this context-sensitive emission probability as

$$e(x_j \mid x_i, y_i, y_j) = P\{x_j \text{ is emitted at } y_j = C_n, \text{ given that } x_i \text{ was emitted at } y_i = P_n\}. \quad (22)$$

Note that by combining the emission probability $e(x_i \mid y_i)$ at a pairwise-emission state $y_i = P_n$ and the emission probability $e(x_j \mid x_i, y_i, y_j)$ at the corresponding context-sensitive state $y_j = C_n$, we obtain the joint emission probability of x_i and x_j

$$P\{x_i, x_j \mid y_i, y_j\} = P\{x_i \mid y_i\} P\{x_j \mid x_i, y_i, y_j\} = e(x_i \mid y_j) e(x_j \mid x_i, y_i, y_j), \quad (23)$$

where we used the fact that x_i is independent of y_j. This clearly shows that we can describe long-range pairwise symbol correlations by using a pair of P_n and C_n, and then specifying their emission probabilities. Since a given pairwise-emission state P_n and its corresponding context-sensitive state C_n work together to describe the symbol correlations, these states always exist in pairs, and a separate memory is allocated to each state pair (P_n, C_n). As we need the contextual information to adjust the emission probabilities at a context-sensitive state, the transition probabilities in the model are adjusted such that we never enter a context-sensitive state if the associated memory is empty [27].

Using context-sensitive HMMs, we can easily describe any kind pairwise symbol correlations by arranging the pairwise emission states P_n and the corresponding context-sensitive states C_n accordingly. As a simple example, let us consider a csHMM that generates *only* symmetric sequences, or palindromes. Such an example is shown in Fig. (**4**).

The model has three states, a pair of pairwise-emission state P_1 and context-sensitive state C_1, and one single-emission state S_1. In this example, the state pair (P_1, C_1) uses a stack, and the two states work together to model the symbol correlations that are induced by the symmetry of the sequence. Initially, the csHMM enters the pairwise-emission

[1]Although different types of memories (stacks, queues, etc.) can be used with csHMMs, it is convenient to use stacks for modeling RNAs.

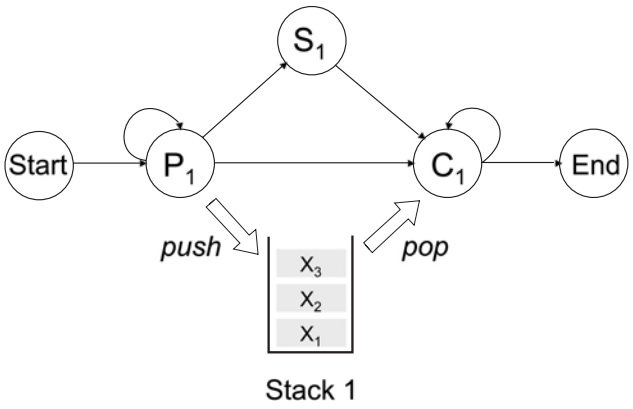

Fig. (4). A context-sensitive HMM that generates only symmetric sequences, or palindromes.

state P_1 and emits one or more symbols. The symbols emitted at P_1 are stored in the stack. When we enter C_1, we first retrieve a symbol from the top of the stack. Based on this symbol, the emission probabilities of C_1 are adjusted such that it emits an identical symbol with probability 1. Transition probabilities of C_1 are adjusted such that it makes a transition to itself until the stack becomes empty. Once the stack becomes empty, the csHMM terminates. In this way, the csHMM shown in Fig. (**4**) generates only palindromes that take one of the following forms

$$\mathbf{x}_e = x_1 x_2 \ldots x_N x_N \ldots x_2 x_1 \ (\textit{even length})$$

$$\mathbf{x}_o = x_1 x_2 \ldots x_N x_{N+1} x_N \ldots x_2 x_1 \ (\textit{odd length}).$$

The underlying state sequences for $\mathbf{x}_e$ and $\mathbf{x}_o$ will be

$$\mathbf{y}_e = \underbrace{P_1 \ldots P_1}_{N \text{ states}} \underbrace{C_1 \ldots C_1}_{N \text{ states}} \quad \textit{and} \quad \mathbf{y}_o = \underbrace{P_1 \ldots P_1}_{N \text{ states}} S_1 \underbrace{C_1 \ldots C_1}_{N \text{ states}},$$

respectively. Note that the single-emission state S_1 is only used to generate the symbol located in the center of a palindrome with odd length, since this symbol is not correlated to any other symbols.

This example clearly shows how we can represent pairwise correlations using a csHMM. When modeling RNAs with conserved base-pairs, we can arrange P_n and C_n based on the positions of the base-pairs, and adjust the emission probabilities at C_n such that they emit the bases that are complementary to the bases emitted at the corresponding P_n. By adjusting the context-sensitive emission probabilities $e(x_j \mid x_i, y_i = P_n, y_j = C_n)$, we can model any kind of base-pairs including non-canonical pairs. Considering that the widely used stochastic context-free grammars can model only nested base-pairs, hence no pseudoknots, the increased modeling capability and the ease of representing any kind of base-paired structures are important advantages of context-sensitive HMMs [9, 61].

5.2. Profile Context-Sensitive HMMs

Suppose we have a multiple alignment of relevant RNA sequences. How can we build a probabilistic model to represent the RNA profile, or the important features in the given RNA alignment? Due to the conservation of secondary structure, multiple RNA alignments often display column-wise correlations. When modeling an RNA profile, it is important to reflect these correlations in the model, along with the conserved sequence information. The *profile context-sensitive HMM (profile-csHMM)* provides a convenient probabilistic framework that can be used for this purpose [9, 65]. Profile-csHMMs are a subclass of context-sensitive HMMs, whose structure is similar to that of profile-HMMs. As it is relatively simple to construct a profile-HMM from a protein or DNA sequence alignment, it is rather straightforward to build a profile-csHMM based on a multiple RNA alignment with structural annotation.

Like conventional profile-HMMs, profile-csHMMs also repetitively use *match states* M_k, *insert states* I_k, and *delete states* D_k to model symbol matches, symbol insertions, and symbol deletions, respectively. The main difference between a profile-HMM and a profile-csHMM is that the profile-csHMM can have three different types of match states. As we have seen in Sec. 5.1, context-sensitive HMMs use three different types of states, where the single-emission states S_n are used to represent the symbols that are not directly correlated to other symbols, while the pairwise-emission states P_n and the context-sensitive states C_n are used together to describe pairwise symbol correlations. In a profile-csHMM, each M_k can choose from these three types of states. Therefore, we can have *single-emission match states*, *pairwise-emission match states*, and *context-sensitive match states*. Single-emission match states are used to represent the columns that are not involved in base-pairing. The pairwise correlations between columns, induced by conserved base-pairs, can be represented by using pairwise-emission match states and the corresponding context-sensitive match states.

As an example, let us assume that we want to construct a profile-csHMM for the alignment shown in Fig. (**5a**). Since the alignment has five columns, we need five match states to represent the sequence profile. There exist two base-pairs in the consensus RNA structure, where the bases in the first column form base-pairs with those in the fourth column, and the bases in the second column form base-pairs with those in the fifth column. In order to describe the correlation between the first and the fourth columns, we use a pairwise-emission state for the first match state M_1 and the corresponding context-sensitive state for the fourth match state M_4. Similarly, we use a pairwise-emission state for M_2 and the corresponding context-sensitive state for M_5. We use a single-emission state for the third match state M_3, since the third column is not involved in base-pairing. By interconnecting the five match states $M_1, M_2, \ldots, M_5$, we

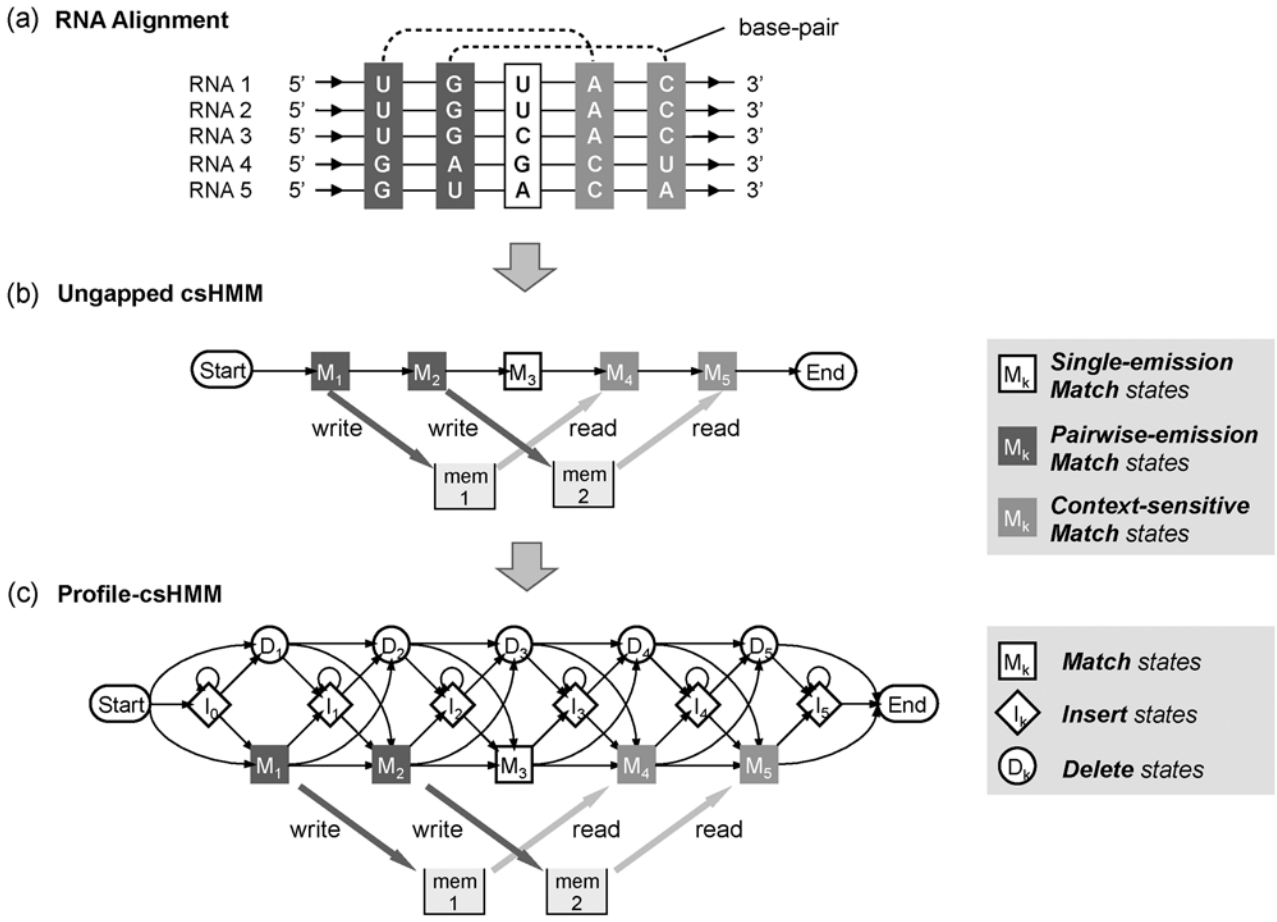

Fig. (5). Constructing a profile-csHMM from a multiple RNA sequence alignment. (**a**) Example of an RNA sequence alignment. The consensus RNA structure has two base-pairs. (**b**) An ungapped csHMM constructed from the given alignment. (**c**) The final profile-csHMM that can handle symbol matches, insertions, and deletions.

obtain an *ungapped* csHMM for the given alignment, as shown in Fig. (**5b**). Finally, we add insert states I_k and delete states D_k to the ungapped model to obtain the final profile-csHMM. Since the inserted bases are not correlated to other bases, we use a single-emission state for each I_k. As in profile-HMMs, the delete states D_k are non-emitting states, and they are simply used to interconnect the neighboring states.

As illustrated in this example, profile-csHMMs provide a convenient tool of modeling RNA profiles. Profile-csHMMs can represent *any* kind of base-pairs by appropriately arranging the pairwise-emission match states and the context-sensitive match states. Due to the increased descriptive capability, algorithms for traditional HMMs (e.g., the Viterbi algorithm) cannot be directly used for profile-csHMMs. However, we can generalize these algorithms so that they can be used with profile-csHMMs. For example, the *sequential component adjoining (SCA)* algorithm [9], which is a generalization of the Viterbi algorithm, provides a systematic way of finding the optimal state sequence in a profile-csHMM.

5.3. Hidden Markov Models in RNA Sequence Analysis

Profile-csHMMs can be used for finding structural alignment of RNAs and performing RNA similarity searches [9, 66]. In [9], the profile-csHMM has been used to find the optimal alignment between a folded RNA (and RNA with a known secondary structure) and an unfolded RNA (an RNA whose folding structure is not known). To find the structural alignment between the two RNAs, we first construct a profile-csHMM to represent the folded RNA. The parameters of the profile-csHMM is chosen according to the scoring scheme proposed in [67]. Based on this model, we use the SCA algorithm to find the optimal state sequence that maximizes the observation probability of the unfolded RNA sequence. The optimal alignment between the two RNAs can be unambiguously determined from the predicted state sequence. Furthermore, we can infer the secondary structure of the unfolded RNA based on the alignment. Theoretically, the profile-csHMM based RNA structural alignment method can handle any kind of pseudoknots. The current implementation of the algorithm [9] can align any RNAs in the Rivas&Eddy class [68] that includes most of the known RNAs [69]. We may use this structural alignment approach for building RNA similarity search tools.

One practical problem that frequently arises in RNA sequence analysis is the high computational complexity. As RNA alignment algorithms have to deal with complicated base-pair correlations, they require significantly more computations compared to sequence-based alignment algorithms. For example, the *Cocke-Younger-Kasami (CYK) algorithm* [3], which is the SCFG analogue of the Viterbi algorithm for HMMs, has a complexity of $O(L^3)$, where L is the length of the RNA to be aligned. Considering that the computational complexity of the Viterbi algorithm increases only linearly with the sequence length, this is a significant increase. The complexity of a simultaneous RNA folding (structure prediction) and alignment algorithm [70] is even higher, and they need $O(L^{3N})$ computations for aligning N RNAs of length L. These algorithms do not consider pseudoknots, and if we allow pseudoknots, the complexity will increase further. The high computational cost often limits the utility of many RNA sequence analysis algorithms, especially when the RNA of interest is long.

To overcome this problem, various heuristics have been developed to expedite RNA alignment and RNA search algorithms. For example, profile-HMM based prescreening filters [11, 71] have been proposed to improve the speed of RNA searches based on *covariance models (CMs)*. Covariance models can be viewed as profile-SCFGs that have a special structure useful for modeling RNA families [3, 63]. In this prescreening approach [11, 71], we first construct a profile-HMM based on the CM that is to be used in the homology search. Note that the resulting profile-HMM conveys only the consensus sequence information of the RNA family represented by the given CM. This profile-HMM is then used to prescreen the genome database to filter out the sequences that are not likely to be annotated as homologues by this CM. The complex CM is run only on the remaining sequences, thereby reducing the average computational cost. It has been demonstrated that using profile-HMM prescreening filters can make the search hundreds of times faster at no (or only a slight) loss of accuracy. A similar approach can be used to speed up profile-csHMM based RNA searches [72].

There also exist a number of methods to improve the speed of simultaneous RNA folding and alignment algorithms [10, 73]. For example, Consan implements a constrained version of the pairwise RNA structure prediction and alignment algorithm based on *pair stochastic context-free grammars (pair-SCFGs)* [73]. It assumes the knowledge of a few confidently aligned base position, called `pins', which are fixed during the alignment process to reduce the overall complexity. These pins are chosen based on the posterior alignment probabilities that are computed using a pair-HMM. A recent version of another pairwise folding and alignment algorithm called Dynalign [10] also employs alignment constraints to improve its efficiency. Dynalign also uses a pair-HMM to compute the posterior alignment and insertion probabilities, which are added to obtain the so-called co-incidence probabilities. We estimate the set of alignable base positions by thresholding the co-incidence probabilities, and this set is subsequently used to constrain the pairwise RNA alignment. It has been shown that employing these alignment constraints can significantly reduce the computational and memory requirements without degrading the structure prediction accuracy [10, 73].

6. CONCLUDING REMARKS

Hidden Markov models have become one of the most widely used tools in biological sequence analysis. In this paper, we reviewed several different types of HMMs and their applications in molecular biology. It has to be noted that this review is by no means exhaustive, and that there still exist many other types of HMMs and an even larger number of sequence analysis problems that have benefited from HMMs. Hidden Markov models provide a sound mathematical framework for modeling and analyzing biological sequences, and we expect that their importance in molecular biology as well as the range of their applications will grow only further.

REFERENCES

[1] Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **1989**, *77*, 257-286.

[2] Munch, K.; Krogh, A. Automatic generation of gene finders for eukaryotic species. *BMC Bioinform.,* **2006**, *7*, 263.

[3] Durbin, R.; Eddy, S.; Krogh, A.; Mitchison, G. *Biological Sequence Analysis*, Cambridge University Press, Cambridge, UK, **1998**.

[4] Pachter, L.; Alexandersson, M.; Cawley, S. Applications of generalized pair hidden Markov models to alignment and gene finding problems. *J. Comput. Biol.*, **2002**, *9*, 389-399.

[5] Liang, K. C.; Wang, X.; Anastassiou, D. Bayesian basecalling for DNA sequence analysis using hidden Markov models. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **2007**, *4*, 430-440.

[6] Lottaz, C.; Iseli, C.; Jongeneel, C. V.; Bucher, P. Modeling sequencing errors by combining Hidden Markov models. *Bioinformatics*, **2003**, *19*(Suppl 2), i103-i112.

[7] Won, K. J.; Hamelryck, T.; Prgel-Bennett, A.; Krogh, A. An evolutionary method for learning HMM structure: prediction of protein secondary structure. *BMC Bioinform.*, **2007**, *8*, 357.

[8] Zhang, S.; Borovok, I.; Aharonowitz, Y.; Sharan, R.; Bafna, V. A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements. *Bioinformatics*, **2006**, *22*, e557-e565.

[9] Yoon, B.-J.; Vaidyanathan, P. P. Structural alignment of RNAs using profile-csHMMs and its application to RNA homology search: Overview and new results. *IEEE Trans. Automat. Contr. (Joint Special Issue on Systems Biology with IEEE Transactions on Circuits and Systems: Part-I)*, **2008**, *53*, 10-25.

[10] Harmanci, A. O.; Sharma, G.; Mathews, D. H. Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign. *BMC Bioinform.*, **2007**, *8*, 130.

[11] Weinberg, Z.; Ruzzo, W. L. Sequence-based heuristics for faster annotation of non-coding RNA families. *Bioinformatics*, **2006**, *22*, 35-39.

[12] Borodovsky, M.; McIninch, J. GENMARK: parallel gene recognition for both DNA strands. *Comput. Chem.*, **1993**, *17*, 123-133.

[13] Trifonov, E. N.; Sussman, J. L. The pitch of chromatin DNA is reflected in its nucleotide sequence. *Proc. Natl. Acad. Sci. USA*, **1980**, *77*, 3816-3820.

[14] Forney, G. D. The Viterbi algorithm. *Proc. IEEE*, **1973**, *61*, 268-278.

[15] Viterbi, A. J. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Inform. Theory*, **1967**, *13*, 260-269.

[16] Baum, L. E.; Petrie, T.; Soules, G.; Weiss, N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.*, **1970**, *41*, 164-171.

[17] Capp´e, O.; Moulines, E.; Ryden, T. *Inference in Hidden Markov Models*, Springer, **2005**.

[18] Levinson, S. E.; Rabiner, L. R.; Sondhi, M. M. An introduction to the application of the theory of probabilistic functions of a Markov

process to automatic speech recognition. *Bell Syst. Tech. J.*, **1983**, *62*, 1035-1074.

[19] Tanner, M. A. *Tools for Statistical Inference*, Springer, **1993**.

[20] Doucet, A.; Godsill, S.; Robert, C. P. Marginal maximum a posteriori estimation using Markov chain Monte Carlo. *Stat. Comput.*, **2002**, *12*, 77-84.

[21] Gaetan, C.; Yao, J.-F. A multiple-imputation Metropolis version of the EM algorithm. *Biometrika*, **2003**, *90*, 643-654.

[22] Gilks, W. R.; Richardson, S.; Spiegelhalter, D. J. *Markov Chain Monte Carlo in Practice* Landon, Chapman and Hall **1996**.

[23] Robert, C. P.; Celeux, G.; Diebolt, J. Bayesian estimation of hidden Markov chains: A stochastic implementation. *Stat. Probab. Lett.*, **1993**, *16*, 77-83.

[24] Eddy, S. R. Profile hidden Markov models. *Bioinformatics*, **1998**, *14*, 755-763.

[25] Krogh, A.; Brown, M.; Mian, I. S.; Sj¨olander, K.; Haussler, D. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, **1994**, *235*, 1501-1531.

[26] Kent, W.; Zahler, A. Conservation, regulation, synteny, and introns in a large-scale *C. briggsae-C. elegans* genomic alignment. *Genome Res.*, **2000**, *10*, 1115-1125.

[27] Yoon, B.-J.; Vaidyanathan, P.P. Context-sensitive hidden Markov models for modeling long-range dependencies in symbol sequences. *IEEE Trans. Signal Processing*, **2006**, *54*, 4169-4184.

[28] Ahola, V.; Aittokallio, T.; Uusipaikka, E.; Vihinen, M. Efficient estimation of emission probabilities in profile hidden Markov models. *Bioinformatics*, **2003**, *19*, 2359-2368.

[29] Bernardes, J. S.; D´avila, A. M.; Costa, V. S.; Zaverucha, G. Improving model construction of profile HMMs for remote homology detection through structural alignment. *BMC Bioinform.*, **2007**, *8*, 435.

[30] Srivastava, P. K.; Desai, D. K.; Nandi, S.; Lynn, A. M. HMM-ModE-improved classification using profile hidden Markov models by optimising the discrimination threshold and modifying emission probabilities with negative training sequences. *BMC Bioinform.*, **2007**, *8*, 104.

[31] Wistrand, M.; Sonnhammer, E. L. Improving profile HMM discrimination by adapting transition probabilities. *J. Mol. Biol.*, **2004**, *338*, 847-854.

[32] Wistrand, M.; Sonnhammer, E. L. Improved profile HMM performance by assessment of critical algorithmic features in SAM and HMMER. *BMC Bioinform.*, **2005**, *6*, 99.

[33] Baldi, P.; Chauvin, Y.; Hunkapiller, T.; McClure, M. A. Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA*, **1994**, *91*, 1059-1063.

[34] Hughey, R.; Krogh, A. SAM: Sequence alignment and modeling software system. *Technical Report*, UCSC-CRL95-7, University of California, Santa Cruz, CA, **1995**.

[35] Karplus, K.; Barrett, C.; Hughey, R. Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **1998**, *14*, 846-856.

[36] Hulo, N.; Bairoch, A.; Bulliard, V.; Cerutti, L.; Cuche, B.; De Castro, E.; Lachaize, C.; Langendijk-Genevaux, P. S.; Sigrist, C. J. A. The 20 years of PROSITE. *Nucleic Acids Res.*, **2008**, *36 (Database issue)*, D245-D249.

[37] Sigrist, C. J. A.; Cerutti, L.; Hulo, N.; Gattiker, A.; Falquet, L.; Pagni, M.; Bairoch, A.; Bucher, P. PROSITE: a documented database using patterns and profiles as motif descriptors. *Brief. Bioinform.*, **2002**, *3*, 265-274.

[38] Finn, R. D.; Tate, J.; Mistry, J.; Coggill, P. C.; Sammut, S. J.; Hotz, H.-R.; Ceric, G.; Forslund, K.; Eddy, S. R.; Sonnhammer, E. L. L.; Bateman, A. The Pfam protein families database. *Nucleic Acids Res.*, **2008**, *36 (Database issue)*, D281-D288.

[39] Sonnhammer, E. L. L.; Eddy, S. R.; Birney, E.; Bateman, A.; Durbin, R. Pfam: Multiple sequence alignments and HMM-profiles of protein domains. *Nuclei Acids Res.*, **1997**, *26*, 320-322.

[40] Edgar, R. C.; Sj¨olander, K. COACH: profile-profile alignment of protein families using hidden Markov models *Bioinformatics*, **2004**, *20*, 1309-1318.

[41] Madera, M. Profile Comparer (PRC): a program for scoring and aligning profile hidden Markov models. *Bioinformatics*, **2008**, *24*, 2630-2631.

[42] Söding, J. Protein homology detection by HMM-HMM comparison. *Bioinformatics*, **2005**, *21*, 951-960.

[43] Di Francesco, V.; Garnier, J.; Munson, P. J. Protein topology recognition from secondary structure sequences: Application of the hidden Markov models to the alpha class proteins. *J. Mol. Biol.*, **1997**, *267*, 446-463.

[44] Di Francesco, V.; Munson, P. J.; Garnier, J. FORESST: fold recognition from secondary structure predictions of proteins. *Bioinformatics*, **1999**, *15*, 131-140.

[45] Pl¨otz, T.; Fink, G. A. Robust remote homology detection by feature based Profile Hidden Markov Models. *Stat. Appl. Genet. Mol. Biol.*, **2005**, *4*, 21.

[46] Kawashima, S.; Kanehisa, M. AAindex: Amino acid index database. *Nucleic Acids Res.*, **2000**, *28*, 374.

[47] Schultz, A. K.; Zhang, M.; Leitner, T.; Kuiken, C.; Korber, B.; Morgenstern, B.; Stanke, M. A jumping profile Hidden Markov Model and applications to recombination sites in HIV and HCV genomes. *BMC Bioinform.*, **2006**, *7*, 265.

[48] Knudsen, B.; Miyamoto, M. M. Sequence alignments and pair hidden Markov models using evolutionary history. *J. Mol. Biol.*, **2003**, *333*, 453-460.

[49] Wang, J.; Keightley, P. D.; Johnson, T. MCALIGN2: faster, accurate global pairwise alignment of non-coding DNA sequences based on explicit models of indel evolution. *BMC Bioinform.*, **2006**, *7*, 292.

[50] Do, C. B.; Mahabhashyam, M. S.; Brudno, M.; Batzoglou, S. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **2005**, *15*, 330-340.

[51] L¨oytynoja, A.; Milinkovitch, M. C. A hidden Markov model for progressive multiple alignment. *Bioinformatics*, **2003**, *19*, 1505-1513.

[52] L¨oytynoja, A.; Goldman, N. An algorithm for progressive multiple alignment of sequences with insertions. *Proc. Natl. Acad. Sci. USA*, **2005**, *102*, 10557-10562.

[53] Feng, D. F.; Doolittle, R. F. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, **1987**, *25*, 351-360.

[54] Thompson, J. D.; Higgins, D. G.; Gibson, T. J. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **1994**, *22*, 4673-4680.

[55] Alexandersson, M.; Cawley, S.; Pachter, L. SLAM: cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res.*, **2003**, *13*, 496-502.

[56] Arumugam, M.; Wei, C.; Brown, R. H.; Brent, M. R. Pairagon+N-SCAN EST: a model-based gene annotation pipeline. *Genome Biol.*, **2006**, *7*(Suppl 1), S5.1-10.

[57] Majoros, W. H.; Pertea, M.; Salzberg, S. L. Efficient implementation of a generalized pair hidden Markov model for comparative gene finding. *Bioinformatics*, **2005**, *21*, 1782-1788.

[58] Meyer, I. M.; Durbin, R. Comparative ab initio prediction of gene structures using pair HMMs. *Bioinformatics*, **2002**, *18*, 1309-1318.

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

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

[61] Yoon, B.-J.; Vaidyanathan, P.P. Computational identification and analysis of noncoding RNAs -Unearthing the buried treasures in the genome. *IEEE Signal Processing Mag.*, **2007**, *24*, 64-74.

[62] Sakakibara, Y.; Brown, M.; Hughey, M.; Mian, I. S.; Sjölander, K.; Underwood, R. C.; Haussler, D. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Res.*, **1994**, *22*, 5112-5120.

[63] Eddy, S. R.; Durbin, R. RNA sequence analysis using covariance models. *Nucleic Acids Res.*, **1994**, *22*, 2079- 2088.

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

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

[66] Yoon, B.-J. Effective annotation of noncoding RNA families using profile context-sensitive HMMs. *Proc. IEEE International Symposium on Communications, Control and Signal Processing (ISCCSP)*, St. Julians, Malta, **2008**.

[67] Gorodkin, J.; Heyer, L. J.; Stormo, G. D. Finding the most significant common sequence an structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, **1997**, *25*, 3724-3732.

[68] Rivas, E.; Eddy, S. R. The language of RNA: a formal grammar that includes pseudoknots. *Bioinformatics*, **2000**, *16*, 334-340.

[69] Condon, A.; Davy, B.; Rastegari, B.; Zhao, S.; Tarrant, F. Classifying RNA Pseudoknotted Structures. *Theor. Comput. Sci.*, **2004**, *320*, 35-50.

[70] Sankoff, D. Simultaneous solution of the RNA folding, alignment, and protosequence problems. *SIAM J. Appl. Math.*, **1985**, *45*, 810-825.

[71] Weinberg, Z.; Ruzzo, W. L. Faster genome annotation of non-coding RNA families without loss of accuracy. *Proc. 8th RECOMB*, **2004**, 243-251.

[72] Yoon, B.-J.; Vaidyanathan, P.P. 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, **2007**.

[73] Dowell, R. D.; Eddy, S. R. Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinform.*, **2006**, *7*, 400.