PROFILE CONTEXT-SENSITIVE HMMS FOR PROBABILISTIC MODELING OF SEQUENCES WITH COMPLEX CORRELATIONS

Byung-Jun Yoon and P. P. Vaidyanathan

Dept. of Electrical Engineering
California Institute of Technology, Pasadena, CA 91125, USA
E-mail: bjyoon@caltech.edu, ppvnath@systems.caltech.edu

ABSTRACT

The profile hidden Markov model is a specific type of HMM that is well suited for describing the common features of a set of related sequences. It has been extensively used in computational biology, where it is still one of the most popular tools. In this paper, we propose a new model called the profile context-sensitive HMM. Unlike traditional profile-HMMs, the proposed model is capable of describing complex long-range correlations between distant symbols in a consensus sequence. We also introduce a general algorithm that can be used for finding the optimal state-sequence of an observed symbol sequence based on the given profile-csHMM. The proposed model has an important application in RNA sequence analysis, especially in modeling and analyzing RNA pseudoknots.

1. INTRODUCTION

The profile hidden Markov model (profile-HMM) [1, 2] is a specific type of HMM that is well suited for describing the key motives and common features of a set of symbol sequences that are closely related to each other. Generally, these sequences can be categorized into the same class according to certain criteria. For example, they may represent different pronunciations of the same word, or different protein-coding genes that give rise to proteins with similar biological functions. A typical way of constructing a profile-HMM begins with finding a multiple alignment of the given sequences. Once the alignment is obtained, the profile-HMM is constructed such that it effectively represents the consensus sequence of the alignment. The aligned sequences are also used for training the HMM, where the probabilities can be obtained by computing the frequencies of all emissions and transitions at each state. This is usually followed by an EM-type parameter optimization in order to maximize the overall observation probability of the training sequences. The HMM obtained in this manner can be used for searching similar regions in a database, or finding the best alignment between a new sequence and the consensus sequence.

Due to the ease of model construction based on multiple alignments as well as its efficiency in capturing short-term dependencies between adjacent symbols, the profile-HMM has been extensively used in biological sequence analysis [2]. It has been especially popular in gene-identification, where the profile-HMM can be used for describing the consensus sequence of certain genes and finding similar regions in novel DNA sequences that have not been annotated yet. In fact, this approach has been quite successful, and many state-of-the-art protein-coding gene-finders are built on profile-HMMs.

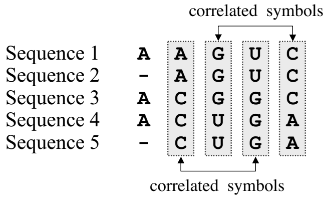

Fig. 1. An example of a multiple alignment of symbol sequences.

One significant limitation of the profile-HMM is the fact that it cannot effectively describe correlations between symbols that are distant from each other. Therefore, any long-range correlations that exist in the consensus sequence get completely lost when the sequence is modeled using a profile-HMM. This problem can be avoided if we represent the profile of the consensus sequence using a context-sensitive HMM (csHMM) [3]. The csHMM has variable probabilities that depend on the context, which greatly increase the overall descriptive power of the HMM.

In this paper, we propose a new model called the *profile context-sensitive HMM* (profile-csHMM) that can be used for constructing a probabilistic profile of related sequences with long-range correlations. The proposed model is based on the concept of context-sensitive HMMs [3, 4], and it is capable of describing *any* kind of pairwise dependencies between distant symbols. We also introduce an algorithm that can be used for finding the optimal state sequence of an observed symbol sequence, based on a profile-csHMM. To the best of our knowledge, this is the first model that is capable of modeling and recognizing any kind of *RNA pseudoknots* (RNA sequences with crossing correlations [2]).

2. THE PROFILE CONTEXT-SENSITIVE HMM

Let us consider the symbol sequences shown in Fig. 1. The consensus sequence obtained from the alignment of these sequences consists of five symbols, where the first symbol is a A (or a gap denoted by '$-$'), the second symbol is either A or C, and so on. One interesting property that can be noticed in Fig. 1 is that the second and the fourth symbols (and also the third and the fifth symbols) are strongly correlated. For example, an A in the second position is always followed by a U in the fourth position, and similarly, a C in the second position is followed by a G in the fourth position. Such correlations are frequently observed in functional RNAs, which typically have strong correlations between bases (represented by symbols A, C, G, U) that arise from the so-called *RNA secondary structure*. In these RNA sequences, the related bases undergo co-variation[1] [2] in such a manner that the

Work supported in parts by the NSF grant CCF-0428326 and the Microsoft Research Graduate Fellowship.

[1] The base A forms a pair with U, and C forms a pair with G. So, if a base (that forms a base-pair) is changed from A to C, the corresponding

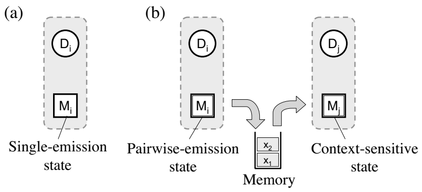

Fig. 2. Basic building blocks of a profile-csHMM.

Watson-Crick complementarity is preserved. In a csHMM, explicit dependencies between distant symbols can be effectively described using a pair of a *pairwise-emission state* and a *context-sensitive state* [4]. For example, in Fig. 1 we may represent the second symbol by a pairwise-emission state, which will store the emitted symbol A (or C) before making a transition to the next state. Now, we represent the fourth symbol using a context-sensitive state, which first accesses the memory once we enter the state. It reads the symbol that was previously emitted at the corresponding pairwise-emission state, and its emission probabilities are adjusted such that the state emits the complementary symbol U (or G). In this way, we can efficiently represent pairwise dependencies between symbols. On the contrary, symbols that are not explicitly correlated with other symbols can be represented using single-emission states.

2.1. Basic Building Blocks

Following the basic idea elaborated just before, it is quite straightforward to construct a profile-csHMM based on a given profile. As in the original profile-HMM [1], we define three different kinds of states, namely, *match states* M_i, *insert states* I_i, and *delete states* D_i. Firstly, an emission at the match state M_i represents the case when a symbol in the observed symbol sequence matches the i-th symbol in the profile. Therefore, if the observation $\mathbf{x} = x_1 x_2 \ldots x_L$ exactly matches the profile, the underlying state sequence will be simply $\mathbf{y} = M_1 M_2 \ldots M_L$. Secondly, the insert state I_i handles insertions of additional symbols in $\mathbf{x}$ that do not exist in the original profile. If the observation $\mathbf{x}$ is longer than the profile of the consensus sequence, a number of symbols in $\mathbf{x}$ will be represented by an insert state I_i. Finally, the delete states D_i deal with gaps that exist in $\mathbf{x}$. In some cases, the observation $\mathbf{x}$ may be shorter than the original profile, which implies that there are symbols in the profile that are missing in $\mathbf{x}$. Such cases are represented by delete states D_i which do not emit any symbols (as they represent gaps).

As every symbol in the consensus sequence should be either "matched" or "deleted", the number of M_i and that of D_i is identical to the length of the consensus sequence. Therefore, we can use a pair of (M_i, D_i) as the basic building block of a profile-csHMM. When a symbol in the profile is not correlated to another symbol, the match state M_i is simply a single-emission state as displayed in Fig. 2 (a). On the contrary, if the i-th symbol and the j-th symbol in the consensus sequence are correlated to each other, we use a pairwise-emission state for M_i and a context-sensitive state for M_j $(i < j)$ as shown in Fig. 2 (b).

2.2. Building a Profile Context-Sensitive HMM

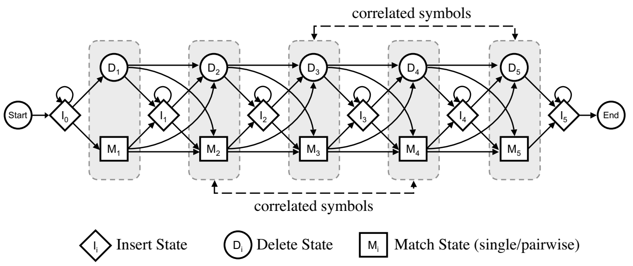

Fig. 3. The profile-csHMM that represents the sequences in Fig. 1.

To obtain the overall model, the two building blocks shown in Fig. 2 are interconnected with additional insert states I_i according to the structure of the consensus sequence. Fig. 3 shows an example of a profile-csHMM that corresponds to the multiple alignment in Fig. 1. As the second symbol and the fourth symbol in the consensus sequence are correlated, a pairwise-emission state is used for M_2 and the corresponding context-sensitive state is used for M_4. Similarly, a pairwise-emission state is used for M_3, where the matching context-sensitive state is used for M_5. The match state M_1 and all insert states $I_0, \ldots, I_5$ are single-emission states.

Once we have obtained the profile-csHMM that reflects the probabilistic profile of the consensus sequence, this model can be used for aligning and scoring new sequences. Fig. 4 shows examples of several observation sequences $\mathbf{x}$ along with the corresponding state sequences $\mathbf{y}$ that are obtained from aligning $\mathbf{x}$ to the profile-csHMM at hand. Although it is more or less straightforward to align the symbol sequences in Fig. 4 to the model illustrated in Fig. 3, finding the best alignment between an observation sequence and a profile-csHMM (which is equivalent to finding the optimal state sequence) can become a daunting task as the length of the sequence and the size of the model increases. Therefore, we need a systematic way of finding the optimal state sequence of an observation $\mathbf{x}$ based on the given profile-csHMM.

When using the traditional profile-HMM, the optimal alignment between the observation and the HMM can be found using a variant of the Viterbi algorithm [2]. However, the Viterbi algorithm cannot be applied to profile context-sensitive HMMs, as there are states with variable probabilities that are dependent on the context. In [4], algorithms were proposed that can be used with csHMMs for a restricted class of symbol correlations (sequences with nested correlations, where correlations do not cross each other). In the following section, we introduce a general algorithm that can be used for finding the optimal alignment of a profile-csHMM with *any kind of pairwise correlations*.

3. FINDING THE OPTIMAL STATE SEQUENCE OF A PROFILE-CSHMM

The basic philosophy that underlies various dynamic programming algorithms that are used for finding the optimal state sequence is as follows. Instead of enumerating all possible state sequences, whose number grows exponentially with the length of the observation sequence, these algorithms try to find the optimal state sequence in a recursive manner. They first find the optimal alignment of short subsequences, and this information is used to find the optimal alignment of longer subsequences. For example, the

A C U G A	M_1 M_2 M_3 M_4 M_5
A A G U U C	M_1 M_2 M_3 M_4 I_4 M_5
- A G U C	D_1 M_2 M_3 M_4 M_5

Fig. 4. (Left) Observation sequence $\mathbf{x}$. (Right) State sequence $\mathbf{y}$ obtained from aligning $\mathbf{x}$ to the profile-csHMM in Fig. 3.

base is also changed from U to G so that the base-pair is maintained.

Viterbi algorithm [5] finds the optimal state sequence by growing the subsequence from left to right, and the CYK algorithm used for parsing stochastic context-free grammars (SCFG) [2, 6] starts from the inside of the observation sequence and proceeds to the outward direction. Although these algorithms cannot be directly used with profile-csHMMs, we can adopt a similar approach for finding the optimal state sequence.

3.1. Notations

Let us first define the variables that are needed in the algorithm. We denote the observed symbol sequence as $\mathbf{x} = x_1 x_2 \ldots x_L$, where L is the length of the sequence. The state sequence of $\mathbf{x}$ is denoted as $\mathbf{y} = y_1 y_2 \ldots y_L$, where y_i is the underlying state of the symbol x_i. At single-emission states and pairwise-emission states, the emission probability of a symbol x at a state v is defined as $e(x|v)$. At context-sensitive states, the emission probability of x_c at the state v is $e(x_c|v, x_p)$, where x_p is the symbol that was previously emitted at the corresponding pairwise-emission state. The transition probability from a state v to w is denoted by $t(v, w)$.

Now, let us define the closed interval of the index n ($1 \leq n \leq L$) as $\mathbf{n}_i = [n_i^\ell,\ n_i^r] = \{n |\ n_i^\ell \leq n \leq n_i^r\}$. For each interval $\mathbf{n}_i$, we define the state-pair $\mathbf{s}_i = (s_i^\ell,\ s_i^r)$, where s_i^ℓ is the hidden state $y_{n_i^\ell}$ at the index n_i^ℓ and s_i^r is the state $y_{n_i^r}$ at n_i^r. The set $\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_I\}$ is an ordered set of non-overlapping closed intervals $\mathbf{n}_i$, where I is the number of the intervals that comprise $\mathcal{N}$. We label each interval $\mathbf{n}_i$ such that it satisfies

$$n_i^r < n_j^\ell \text{ for } i < j. \tag{1}$$

We also define the set of state-pairs $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_I\}$. The set $\mathcal{N}$ will be used for indexing subsequences of $\mathbf{x}$, where the subsequences corresponding to this set are defined as

$$\mathbf{x}(\mathcal{N}) = x_{n_1^\ell} \cdots x_{n_1^r} x_{n_2^\ell} \cdots x_{n_2^r} \cdots x_{n_I^\ell} \cdots x_{n_I^r} \tag{2}$$

$$\mathbf{y}(\mathcal{N}) = y_{n_1^\ell} \cdots y_{n_1^r} y_{n_2^\ell} \cdots y_{n_2^r} \cdots y_{n_I^\ell} \cdots y_{n_I^r}. \tag{3}$$

Finally, let us denote the optimal log-probability of the subsequence $\mathbf{x}(\mathcal{N})$, where the state at either end of each closed interval $\mathbf{n}_i$ is $y_{n_i^\ell} = s_i^\ell$ and $y_{n_i^r} = s_i^r$ ($i = 1, \ldots, I$), as $\alpha(\mathcal{N}, \mathcal{S})$. It is assumed that explicit correlations between symbols in $\mathbf{x}(\mathcal{N})$ are confined inside this subsequence. We also define the variables $\lambda_a(\mathcal{N}, \mathcal{S})$ and $\lambda_b(\mathcal{N}, \mathcal{S})$ that will be used later for tracing back the optimal state sequence $\mathbf{y}^*$.

3.2. Algorithm

Now, the optimal alignment algorithm can be described as follows.

3.2.1. Initialization

(i) For any single-emission state v, and for $1 \leq n \leq L$, we let $\mathcal{N} = \{[n, n]\}$, $\mathcal{S} = \{(v, v)\}$

$$\alpha\Big(\mathcal{N}, \mathcal{S}\Big) = \log e(x_n|v)$$

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

(ii) For any pair (v, w), where $v = M_i$ is a pairwise-emission state and $w = M_j$ is the corresponding context-sensitive state, and for $1 \leq n < m \leq L$, we let $\mathcal{N} = \{[n, n], [m, m]\}$, $\mathcal{S} = \{(v, v), (w, w)\}$

$$\alpha\Big(\mathcal{N}, \mathcal{S}\Big) = \log e(x_n|v) + \log e(x_m|w, x_n)$$

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

(iii) For $v = D_i$ (where M_i is a single-emission state), and for $1 \leq n \leq L$, we let $\mathcal{N} = \{[n, n-1]\}$, $\mathcal{S} = \{(v, v)\}$

$$\alpha\Big(\mathcal{N}, \mathcal{S}\Big) = 0$$

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

(iv) For all (i, j) where M_i and M_j are paired states, and for $1 \leq n < m \leq L$, we let $\mathcal{N} = \{[n, n-1], [m, m-1]\}$, $\mathcal{S} = \{(D_i, D_i), (D_j, D_j)\}$

$$\alpha\Big(\mathcal{N}, \mathcal{S}\Big) = 0$$

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

3.2.2. Recursion

During the initialization process, we computed the log-probability for all subsequences of length up to two. Now, these subsequences can be recursively adjoined to obtain the probability of longer sequences 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

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

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

These subsequences $\mathbf{x}(\mathcal{N}_a)$ and $\mathbf{x}(\mathcal{N}_b)$ can be adjoined only if there is no overlapping interval between $\mathcal{N}_a$ and $\mathcal{N}_b$. In this case, we can apply the following adjoining rule

$$\alpha(\mathcal{N}, \mathcal{S}) = \alpha(\mathcal{N}_a, \mathcal{S}_a) + \alpha(\mathcal{N}_b, \mathcal{S}_b)$$

$$\lambda_a(\mathcal{N}, \mathcal{S}) = (\mathcal{N}_a, \mathcal{S}_a),\ \lambda_b(\mathcal{N}, \mathcal{S}) = (\mathcal{N}_b, \mathcal{S}_b).$$

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

$$\mathcal{N} = \mathcal{N}_a \cup \mathcal{N}_b = \{\mathbf{n}_1, \ldots, \mathbf{n}_I\}, \mathcal{S} = \mathcal{S}_a \cup \mathcal{S}_b = \{\mathbf{s}_1, \ldots, \mathbf{s}_I\}$$

where $I = I_a + I_b$ and the intervals $\mathbf{n}_i$ are relabeled such that they satisfy (1) and $\mathbf{s}_i \in \mathcal{S}$ corresponds $\mathbf{n}_i \in \mathcal{N}$.

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

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

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

$$\alpha(\mathcal{N}', \mathcal{S}') = \max_{n_{I-1}^r} \left(\max_{s_{I-1}^r, s_I^\ell} \Big[\alpha(\mathcal{N}, \mathcal{S}) + \log t(s_{I-1}^r, s_I^\ell)\Big] \right)$$

$$(n^*, s_r^*, s_\ell^*) = \operatorname*{arg\,max}_{(n_{I-1}^r, s_{I-1}^r, s_I^\ell)} \Big[\alpha(\mathcal{N}, \mathcal{S}) + \log t(s_{I-1}^r, s_I^\ell)\Big]$$

$$\mathcal{N}^* = \Big\{\mathbf{n}_1, \ldots, \mathbf{n}_{I-2}, [n_{I-1}^\ell, n^*], [n^* + 1, n_I^r]\Big\}$$

$$\mathcal{S}^* = \Big\{\mathbf{s}_1, \ldots, \mathbf{s}_{I-2}, (s_{I-1}^\ell, s_r^*), (s_\ell^*, s_I^r)\Big\}$$

$$\lambda_a(\mathcal{N}', \mathcal{S}') = (\mathcal{N}^*, \mathcal{S}^*),\ \lambda_b(\mathcal{N}', \mathcal{S}') = (\varnothing, \varnothing)$$

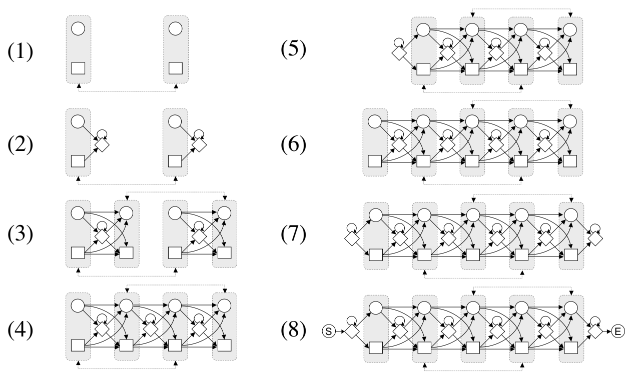

Fig. 5. Adjoining order of subsequences for finding the optimal alignment of the profile-csHMM in Fig. 3.

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 $\alpha(\mathcal{N}', \mathcal{S}')$ of the updated sets $\mathcal{N}'$ and $\mathcal{S}'$. □

Given the two rules above, the immediate question is how we should apply these rules to obtain the probability of the optimal state sequence. In fact, how the components are adjoined and in which order the adjoining rules are applied have a crucial impact on the overall complexity of the algorithm. This can be easily seen from the second adjoining rule. When applying this rule, the computational cost is of the order $O(M^2 L)$, where M is the number of states in the model. Every time the number of "junctions" between $\mathcal{N}_1$ and $\mathcal{N}_2$ increase, the computational cost for adjoining them will be increased by and order of $M^2 L$. For this reason, it is important to apply these rules in an efficient manner to minimize the computational cost. The rule of thumb is to adjoin the components in a way that can take care of all kinds of correlations in the given model, while keeping the number of closed intervals in any $\mathcal{N}$ used in the adjoining process as small as possible. For example, if there is no explicit dependencies between symbols as in a traditional HMM, we can simply use a single closed interval $\mathcal{N} = \{[n^\ell,\ n^r]\}$ with $n^\ell = 0$ fixed, and attach other one-symbol subsequences to its right end[2]. Similarly, when only nested correlations are considered, the adjoining rules can be applied using only one closed interval $\mathcal{N} = \{[n^\ell,\ n^r]\}$ with variable n^ℓ and n^r. Fig. 5 shows *an* example in which order $\alpha(\mathcal{N}, \mathcal{S})$ can be computed to find the optimal alignment.

3.2.3. *Termination*

We repeat the recursion until we get $\alpha(\mathcal{N}, \mathcal{S})$, where $\mathcal{N} = \{[1, L]\}$ and $\mathcal{S} = \{(s_1^\ell, s_1^r)\}$ for all s_1^ℓ, s_1^r. Now, the log-probability of the optimal state-sequence $\mathbf{y}^*$ can be computed from

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

3.2.4. *Trace-Back*

Now that we have computed the maximum probability $P(\mathbf{x}, \mathbf{y}^*)$, we can trace-back the algorithm to find the optimal state sequence $\mathbf{y}^*$ that gave rise to this probability. For notational convenience, let us define $\lambda_t = (\mathcal{N}, \mathcal{S})$. The trace-back procedure can be described as follows.

STEP 1 Let $y_i = 0$ $(i = 1, 2, \ldots, L)$.

STEP 2 Push λ^* onto the stack T.

STEP 3 Pop $\lambda_t = (\mathcal{N}, \mathcal{S})$ from T. If $\lambda_t = (\varnothing, \varnothing)$, goto STEP 6. Otherwise, proceed to STEP 4.

STEP 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 $s_i^\ell = s_i^r$.)

STEP 5 If $\lambda_b(\lambda_t) \neq (\varnothing, \varnothing)$ push $\lambda_b(\lambda_t)$ onto T.

STEP 6 If T is empty, proceed to STEP 7. Otherwise, goto STEP 3.

STEP 7 Let $\mathbf{y}^* = y_1 y_2 \ldots y_L$ and terminate.

At the end of the trace-back procedure, we can obtain the optimal state sequence $\mathbf{y}^*$ that maximizes the probability of observing $\mathbf{x}$ based on the model at hand.

4. CONCLUDING REMARKS

As shown in this paper, the proposed profile-csHMM has a greater descriptive power than many existing models including the profile-HMM and the SCFG. The profile-csHMM is capable of modeling *any* kind of pairwise-dependencies between symbols, by interconnecting the basic building blocks in Fig. 2 according to the structure of the consensus sequence. We also proposed a general algorithm for finding the optimal state sequence of a profile-csHMM. Widely used optimal alignment algorithms such as the Viterbi algorithm [5] for HMMs and the CYK algorithm [2, 6] for SCFGs can be viewed as special cases of the algorithm proposed in this paper. To the best of our knowledge, the profile-csHMM is the first model that can be used for constructing a probabilistic profile of any kind of pseudoknots, hence it can serve as an effective framework for computational RNA sequence analysis.

5. REFERENCES

[1] S. R. Eddy, "Hidden Markov models", *Current Opinion in Structural Biology*, vol. 6, pp. 361-365, 1996.

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

[3] B.-J. Yoon and P. P. Vaidyanathan, "HMM with auxiliary memory: a new tool for modeling RNA secondary structures", *Proc. 28th Asilomar Conf. on Signals, Systems, and Computers*, Monterey, CA, Nov. 2004.

[4] B.-J. Yoon and P. P. Vaidyanathan, "Context-sensitive hidden Markov models for modeling long-range dependencies in symbol sequences", *IEEE Trans. Signal Proc.*, to appear.

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

[6] K. Lari and S. J. Young, "The estimation of stochastic context-free grammars using the inside-outside algorithm", *Computer Speech and Language*, vol. 4, pp. 35-56, 1990.

[2] In this case, the algorithm becomes identical to the Viterbi algorithm.