



Pair hidden Markov models on tree structures

Yasubumi Sakakibara

Department of Biosciences and Informatics, Keio University, 3-14-1 Hiyoshi,
Kohoku-ku, Yokohama, 223-8522, Japan

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

Motivation: Computationally identifying non-coding RNA regions on the genome has much scope for investigation and is essentially harder than gene-finding problems for protein-coding regions. Since comparative sequence analysis is effective for non-coding RNA detection, efficient computational methods are expected for structural alignments of RNA sequences. On the other hand, Hidden Markov Models (HMMs) have played important roles for modeling and analysing biological sequences. Especially, the concept of Pair HMMs (PHMMs) have been examined extensively as mathematical models for alignments and gene finding.

Results: We propose the pair HMMs on tree structures (PHMMTs), which is an extension of PHMMs defined on alignments of trees and provides a unifying framework and an automata-theoretic model for alignments of trees, structural alignments and pair stochastic context-free grammars. By structural alignment, we mean a pairwise alignment to align an unfolded RNA sequence into an RNA sequence of known secondary structure. First, we extend the notion of PHMMs defined on alignments of 'linear' sequences to pair *stochastic tree automata*, called PHMMTs, defined on alignments of 'trees'. The PHMMTs provide various types of alignments of trees such as affine-gap alignments of trees and an automata-theoretic model for alignment of trees. Second, based on the observation that a secondary structure of RNA can be represented by a tree, we apply PHMMTs to the problem of structural alignments of RNAs. We modify PHMMTs so that it takes as input a pair of a 'linear' sequence and a 'tree' representing a secondary structure of RNA to produce a structural alignment. Further, the PHMMTs with input of a pair of two linear sequences is mathematically equal to the pair stochastic context-free grammars. We demonstrate some computational experiments to show the effectiveness of our method for structural alignments, and discuss a complexity issue of PHMMTs.

Contact: yasu@bio.keio.ac.jp

Keywords: hidden Markov model, tree automaton, structural alignment, stochastic context-free grammar, RNA, secondary structure.

INTRODUCTION

Since there are a rapidly growing number of RNA sequences, structures and families, computational methods for finding non-protein-coding RNA regions on the genome have much scope for investigation (Eddy, 2001; Schattner, 2002). Compared with genefinding problems for protein-coding regions, computationally identifying non-coding RNA regions is essentially harder because non-coding RNA sequences do not have strong statistical signals, and there are as yet no general finding algorithms. Several works (Eddy and Durbin, 1994; Gorodkin *et al.*, 1997; Holmes and Rubin, 2002; Knudsen and Hein, 1999; Mathews and Turner, 2002; Rivas and Eddy, 2001; Sakakibara *et al.*, 1994; Sankoff, 1985) have proposed that comparative sequence analysis is effective for non-coding RNA detection and presented various methods for alignments of RNA sequences. When we compare RNA sequences, their secondary structures play an important role in obtaining precise alignments.

On the other hand, Hidden Markov Models (HMMs), especially Pair HMMs (PHMMs) such as automata-theoretic models (Durbin *et al.*, 1998; Searls and Murphy, 1995), generalized PHMMs (GPHMMs) (Pachter *et al.*, 2002) and pair stochastic context-free grammars (PSCFGs) (Rivas and Eddy, 2001; Holmes and Rubin, 2002), have been extensively studied for modeling and solving alignments of sequences and gene finding. PHMMs are an extension of standard HMMs and can generate an aligned pair of sequences. PHMMs also provide an automata-theoretic interpretation for pairwise alignments of sequences and clearly model affine-gap alignment and any other types of alignments.

In this paper, we propose the pair HMMs on tree structures (PHMMTs), which is an extension of PHMMs defined on alignments of trees and provides a unifying framework and an automata-theoretic model for alignments of trees, structural alignments, and PSCFGs. We extend the notion of pair HMMs defined on alignments of 'linear' sequences to pair stochastic tree automata, called *pair HMMs on tree structures* (PHMMTs), defined on alignments of 'trees'. The notion of *alignment of trees* is introduced by Jiang *et al.* (1995), and our PHMMTs

provide various types of alignments of trees such as affine-gap alignments of trees and an automata-theoretic model for alignment of trees. Next, based on the observation that a secondary structure of RNA can be represented by a tree, we apply PHMMTSs to the problem of structural alignments of RNAs. By the structural alignment, we mean a pairwise alignment to align an unfolded RNA sequence into an RNA sequence of known secondary structure. We modify PHMMTSs so that it takes as input a pair of a ‘linear’ sequence and a ‘tree’ representing a secondary structure of RNA to produce a structural alignment. This is accomplished by introducing a technique for parsing sequences with context-free grammars such as Cocke–Younger–Kasami algorithm (Aho and Ullman, 1972). Further, we show that PHMMTSs with input of a pair of two linear sequences is mathematically equal to PSCFGs.

We demonstrate some computational experiments on two RNA families to show the effectiveness of our PHMMTSs for structural alignment. In our experiments, we randomly choose one RNA sequence annotated with known secondary structure in an RNA family and structurally align all other ‘unfolded’ RNA sequences in the family into the chosen ‘folded’ RNA sequence. We count the fraction of base pairs specified by the trusted alignment that matched in the predictions by our structural alignment algorithm. Experimental results show that for transfer RNA family, our method predicts secondary structures extremely well, and for RNA family of hammerhead ribozyme, the prediction accuracy is around 80% compared with predictions of Covariance Model (Eddy and Durbin, 1994; Griffiths-Jones *et al.*, 2003) which is fully trained and designed specific to the family. One important fact is that our method does not require any training process and predicts secondary structures only based on structural alignment into one single folded RNA sequence.

For the problem of structural alignments of RNA sequences, most comparative works are the stochastic context-free grammars (SCFGs), called *profile* SCFGs, for modeling RNA sequences (Eddy and Durbin, 1994; Sakakibara *et al.*, 1994). One significant difference between profile SCFGs and our PHMMTSs is that the profile-SCFG methods require a training stage and a number of sequences for the training to fit grammars to a family of RNAs. This argument may be understood as similar comparisons between pairwise alignments and profiles and between PHMMs and profile HMMs (Durbin *et al.*, 1998; Eddy, 1998). The computational cost to execute PHMMTSs for structural alignments is theoretically same order as the computational complexity to parse an input sequence with SCFGs. However, the training cost for SCFGs with a number of training sequences is huge, and larger than the parsing cost. In this sense, our experiments illustrate the trade-off between

the computational and resource costs and the prediction accuracy.

PAIR HIDDEN MARKOV MODELS ON TREE STRUCTURES

Alignment of trees

The notion of alignment of sequences has been extended to *alignment of trees* (Jiang *et al.*, 1995). A tree is a rooted, directed, connected acyclic finite graph in which the direct successors of any node are linearly ordered from left to right. The predecessor of a node is called the *parent*, the successor, a *child*. If a node has k children, we can designate them as the first child, the second child, and so on up to k th child. Let θ denote the empty tree and λ denote the null label.

Let T be a tree. *Inserting* a node u into T means that for some node v in T , we make u the parent of a consecutive subsequence of the children of v and then v the parent of u . Let T_1 and T_2 be two labeled trees. An *alignment* of T_1 and T_2 is obtained by first inserting nodes labeled with null λ into T_1 and T_2 such that two resulting trees T'_1 and T'_2 have the same structure, that is, they are identical if the labels are ignored, and then overlaying T'_1 on T'_2 . A score is defined for each pair of labels. The *value* of the alignment is the sum of the scores of all pairs of corresponding labels. An *optimal* alignment is one that minimizes the value over all possible alignments. Jiang *et al.* (1995) gave the recurrence equations to calculate optimal alignments of trees.

Hidden Markov models on tree structures: stochastic tree automata

A Hidden Markov model (HMM) is mathematically equal to stochastic finite automaton on the domain of linear sequences. We extend the notion of HMMs defined on sequences to the one for tree structures (called *HMMs on tree structures*) by introducing stochastic tree automata.

A tree automaton is an extension of finite automata and defined on the domain of trees (Sakakibara, 1992). A Stochastic tree automaton is a stochastic version of a tree automaton where some probability is attached to each state transition and the tree automaton assigns a probability to each tree.

We represent the ‘tree’ T as follows. The nodes in a tree T of size m are numbered from 1 to m according to the preorder where the root node is numbered 1. Let T_1 and T_2 be two labeled trees. We denote the label of node j in tree T_l as $v_l(j)$ and the subtree of T_l rooted at node j as $T_l[j]$. Let i be a node of a tree and have k children. We denote the children of i as i_1, i_2, \dots, i_k , and we call the number of children the *arity* of a node i . In this paper, we only consider the trees where the arity of every node is at most two.

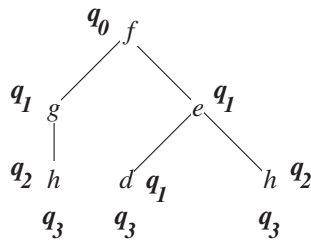


Fig. 1. A state transitions of tree automaton on an input tree.

(Top-down) *Tree automaton* M is defined by a five tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a set of states, Σ is a set of labels, δ is a state transition function from a state and a label to a tuple of states, $q_0 \in Q$ is the initial state, and F is a set of final states. An example of tree automaton M is defined as follows and its state-transition process for a tree is illustrated in Figure 1:

$$M = (Q, \Sigma, \delta, q_0, F) :$$

$$Q = \{q_0, q_1, q_2, q_3\}, \Sigma = \{d, e, f, g, h\}, F = \{q_3\},$$

$$\delta(q_0, f) = (q_1, q_1), \delta(q_1, g) = q_2, \delta(q_1, e) = (q_1, q_2),$$

$$\delta(q_1, d) = q_3, \delta(q_2, h) = q_3.$$

A *Stochastic tree automaton*, called *HMM on tree structures* (HMMTS), associates a probability value with every state transition, accepts trees and assigns a probability to each tree. In this paper, we separately define the state-transition probability and the emission probability of labels. Further, we assume that the probability of state transition from a state to a tuple of states can be calculated by taking the product of all state-transition probabilities from the state to states in the tuple. For example, $\Pr(\delta(q_1, e) = (q_1, q_2))$ is assumed to be equal to the probability $\Pr(q_1|q_1) \cdot \Pr(q_2|q_1) \cdot \Pr(e|q_1)$.

Pair HMMs on tree structures: a unifying framework

Our main contribution in this paper is to propose pair hidden Markov models on tree structures, which is an extension of PHMMs defined on alignments of trees and provides a unifying framework and an automata-theoretic model for alignments of trees, structural alignments and pair stochastic context-free grammars.

Pair HMMs are an extension of standard HMMs and can generate an aligned pair of sequences. PHMMs have been extensively studied as automata-theoretic models for pairwise alignments of sequences (Durbin *et al.*, 1998; Searls and Murphy, 1995). We introduce *pair HMMs on tree structures* (PHMMTSs) which are obtained by modifying a stochastic tree automaton so that it emits a pairwise alignment of trees instead of emitting a single tree.

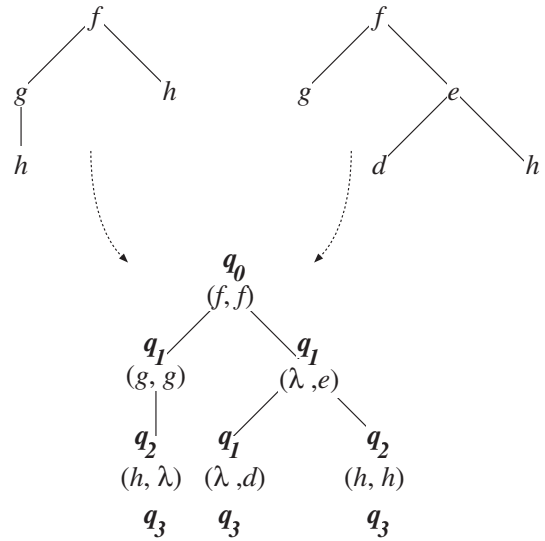


Fig. 2. A pair of two input trees (above), and a state-transition process to accept an alignment of the pair of trees (below). The pair tree automaton for PHMMTS inserts nodes labeled with null λ into both trees such that two resulting trees have the same structure, and accepts an alignment for the trees.

First, the underlying (deterministic) pair tree automaton M which emits a pair of labels is defined by a five tuple $M = (Q, (\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\}), \delta, q_0, F)$, where $(\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\})$ is a set of pairs of labels and the null label λ , and δ is a state transition function: $Q \times ((\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\})) \rightarrow (Q \times \dots \times Q)$. An example of pair tree automaton M is given below and its state-transition process for a pair of tree is illustrated in Figure 2:

$$M = (Q, (\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\}), \delta, q_0, F) :$$

$$Q = \{q_0, q_1, q_2, q_3\}, \Sigma = \{d, e, f, g, h\}, F = \{q_3\},$$

$$\delta(q_0, (f, f)) = (q_1, q_1), \delta(q_1, (g, g)) = q_2,$$

$$\delta(q_1, (\lambda, e)) = (q_1, q_2), \delta(q_1, (\lambda, d)) = q_3,$$

$$\delta(q_2, (h, \lambda)) = q_3, \delta(q_2, (h, h)) = q_3.$$

Second, the PHMMTS based on a pair tree automaton is defined to associate a probability value with every state transition, accept alignments of trees and assign a probability to each alignment. Further, the underlying pair tree automaton is extended to the *non-deterministic* pair tree automaton so that it accepts a set of various alignments for a same input pair of trees. The PHMMTS assigns a probability to each alignment of trees in the set of all possible alignments and can provide the most likely alignment for the pair of trees with highest probability among the alternatives. The most likely alignment can be efficiently computed by using dynamic programming.

The following is the recurrence equations to calculate affine-gap alignments of an input pair of trees T_1 and T_2 based on PHMMTSs with three states, *match state M*, *insertion state I* and *deletion state D*:

$$p^M(T_1[i], T_2[j]) = p_O^M(v_1(i), v_2(j)) \cdot \max_{X, Y \in \{M, I, D\}} \left\{ \begin{array}{l} \delta_{MX} \cdot p^X(T_1[i_1], T_2[j_1]) \\ \quad \cdot \delta_{MY} \cdot p^Y(T_1[i_2], T_2[j_2]), \\ \quad \text{if both } i \text{ and } j \text{ are nodes of arity 2,} \\ \delta_{MX} \cdot p^X(T_1[i_1], T_2[j_1]), \\ \quad \text{if } i \text{ and } j \text{ are of arity 1,} \\ \delta_{MI} \cdot p^I(T_1[i_1], \theta) \\ \quad \cdot \delta_{MX} \cdot p^X(T_1[i_2], T_2[j_1]), \\ \quad \text{if } i \text{ is of arity 2 and } j \text{ is of arity 1,} \\ \delta_{MI} \cdot p^I(T_1[i_2], \theta) \\ \quad \cdot \delta_{MX} \cdot p^X(T_1[i_1], T_2[j_1]), \\ \quad \text{if } i \text{ is of arity 2 and } j \text{ is of arity 1,} \\ \delta_{MD} \cdot p^D(\theta, T_2[j_1]) \\ \quad \cdot \delta_{MX} \cdot p^X(T_1[i_1], T_2[j_2]), \\ \quad \text{if } i \text{ is of arity 1 and } j \text{ is of arity 2,} \\ \delta_{MD} \cdot p^D(\theta, T_2[j_2]) \\ \quad \cdot \delta_{MX} \cdot p^X(T_1[i_1], T_2[j_1]), \\ \quad \text{if } i \text{ is of arity 1 and } j \text{ is of arity 2,} \end{array} \right.$$

$$p^I(T_1[i], T_2[j]) = p_O^I(v_1(i), \lambda) \cdot \max_{X \in \{I, M\}} \left\{ \begin{array}{l} p^I(T_1[i_1], \theta) \cdot \delta_{IX} \cdot p^X(T_1[i_2], T_2[j]), \\ \quad \text{if } i \text{ is of arity 2,} \\ p^I(T_1[i_2], \theta) \cdot \delta_{IX} \cdot p^X(T_1[i_1], T_2[j]), \\ \quad \text{if } i \text{ is of arity 2,} \\ \delta_{IX} \cdot p^X(T_1[i_1], T_2[j]), \\ \quad \text{if } i \text{ is of arity 1,} \end{array} \right.$$

$$p^D(T_1[i], T_2[j]) = p_O^D(\lambda, v_2(j)) \cdot \max_{X \in \{D, M\}} \left\{ \begin{array}{l} p^D(\theta, T_2[j_1]) \cdot \delta_{DX} \cdot p^X(T_1[i], T_2[j_2]), \\ \quad \text{if } j \text{ is of arity 2,} \\ p^D(\theta, T_2[j_2]) \cdot \delta_{DX} \cdot p^X(T_1[i], T_2[j_1]), \\ \quad \text{if } j \text{ is of arity 2,} \\ \delta_{DX} \cdot p^X(T_1[i], T_2[j_1]), \\ \quad \text{if } j \text{ is of arity 1,} \end{array} \right.$$

$$p^X(\theta, \theta) = 1, \quad \text{for } X \in \{M, I, D\},$$

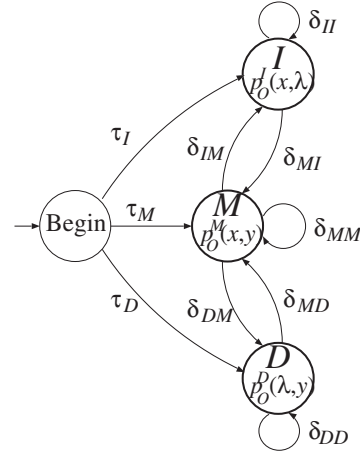


Fig. 3. A state-transition diagram of PHMMTS for affine-gap alignment on tree structures.

where δ_{XY} for $X, Y \in \{M, I, D\}$ denotes the state transition probability from state X to state Y , $p_O^M(v_1(i), v_2(j))$ denotes the emission probability for the pair of labels $v_1(i), v_2(j)$ at the match state M , $p_O^I(v_1(i), \lambda)$ denotes the emission probability at the insertion state to emit a single label $v_1(i)$ for tree T_1 and no label for tree T_2 , and $p_O^D(\lambda, v_2(j))$ denotes the emission probability at the deletion state to emit no label for tree T_1 and a single label $v_2(j)$ for tree T_2 .

A state transition diagram for affine-gap alignments of trees is given in Figure 3. An optimal alignment of trees T_1 and T_2 is obtained by calculating $\max\{\tau_M \cdot p^M(T_1[1], T_2[1]), \tau_I \cdot p^I(T_1[1], T_2[1]), \tau_D \cdot p^D(T_1[1], T_2[1])\}$ for some predefined initial probabilities τ_M, τ_I, τ_D .

The recurrence equations for PHMMTS combined with such state-transition diagram can produce various types of alignments of trees and extend Jiang *et al.* (1995) method for alignments of trees. Next, we will show that PHMMTSs provide a unifying framework for alignments of trees, structural alignments and pair SCFGs.

Structural alignments of RNAs

The second result in this paper is to propose an efficient method for structural alignments between a folded RNA sequence and unfolded RNA sequences based on PHMMTSs, and exhibit the effectiveness of our structural alignment method in some experiments for folding RNA sequences at the next section. By structural alignment, we mean a pairwise alignment to align an unfolded RNA sequence into an RNA sequence of known secondary structure.

The secondary structures of RNAs, which are composed of *stem* (base-pairing), *hairpin*, *bulge*, *interior loop*, and

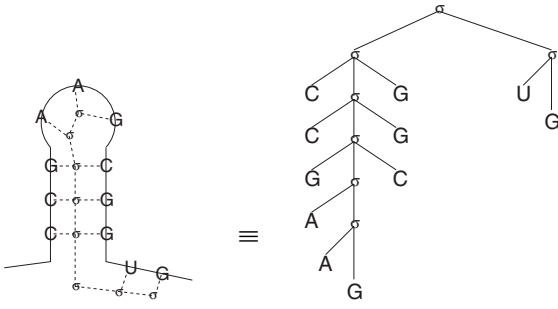


Fig. 4. An RNA secondary structure (left), and a skeletal tree representation for the secondary structure, and an equivalent sequence representation with parentheses inserted (right).

multi-branch, can be conveniently expressed by a special type of tree, called *skeletal tree*, whose internal nodes have no labels or equivalently are labeled by a special symbol ‘ σ ’. More specifically, a folded RNA sequence can be represented by a skeletal tree as follows. Each leaf node is labeled by one of four nucleotides {A, U, G, C} and all internal nodes are labeled by the special symbol σ . The sequence of nucleotides labeled at leaf nodes traced from left to right exactly constitutes the RNA sequence, and the structure of the tree represents its folding structure. A node of the form $\sigma(\sigma, \sigma)$ in a skeletal tree describes branched structures, a node of the form $\sigma(X, \sigma, Y)$ describes base-pairs, and a node of the form $\sigma(X, \sigma)$ or $\sigma(\sigma, X)$ describes unpaired bases, for $X, Y \in \{A, U, G, C\}$. An example of RNA secondary structure and the corresponding skeletal tree are illustrated in Figure 4. Note that a skeletal tree equivalent to an RNA secondary structure is also represented by an RNA sequence with parentheses inserted such as ‘(C(C(GAAGC)G)G)UG’.

Let $w = a_1a_2 \cdots a_n$ be an unfolded RNA sequence of length n and T be a skeletal tree representing a folded RNA sequence of known secondary structure. Let $w[i]$ ($1 \leq i \leq n$) denote the i th symbol a_i in w and $w[i, j]$ ($1 \leq i \leq j \leq n$) denote a substring $a_i a_{i+1} \cdots a_j$ of w . Let ϵ denote the empty sequence.

Here, we present the recurrence equations to calculate optimal structural alignments of an affine-gap model between an unfolded RNA sequence and a folded RNA sequence which is equivalently represented by the corresponding skeletal tree. These recurrence equations are based on the recurrence equations for PHMMTSs shown in the previous section. We modify the recurrence equations for alignments of trees so that the first arguments of the functions p^X ($X \in \{M, I, D\}$) take ‘linear’ sequences as input instead of trees. This is accomplished by introducing a technique for parsing sequences with stochastic context-free grammars such as Cocke–Younger–Kasami

algorithm (Aho and Ullman, 1972).

$$p^M(w[i, k], T[j]) = \max_{\substack{X, Y \in \{M, I, D\} \\ i < l \leq k}} \left\{ \begin{array}{l} p_O^M(\sigma(\sigma, \sigma), v(j)) \\ \quad \cdot \delta_{MX} \cdot p^X(w[i, l-1], T[j_1]) \\ \quad \cdot \delta_{MY} \cdot p^Y(w[l, k], T[j_2]), \\ p_O^M(\sigma(w[i], \sigma, w[k]), v(j)) \cdot \delta_{MX} \\ \quad \cdot p^X(w[i+1, k-1], T[j_1]), \\ p_O^M(\sigma(w[i], \sigma), v(j)) \\ \quad \cdot \delta_{MX} \cdot p^X(w[i+1, k], T[j_1]), \\ p_O^M(\sigma(\sigma, w[k]), v(j)) \\ \quad \cdot \delta_{MX} \cdot p^X(w[i, k-1], T[j_1]). \end{array} \right.$$

$$p^I(w[i, k], T[j]) = \max_{\substack{X \in \{I, M\} \\ i < l \leq k}} \left\{ \begin{array}{l} p_O^I(\sigma(\sigma, \sigma), \lambda) \cdot p^I(w[i, l-1], \theta) \\ \quad \cdot \delta_{IX} \cdot p^X(w[l, k], T[j]), \\ p_O^I(\sigma(\sigma, \sigma), \lambda) \cdot p^I(w[l, k], \theta) \\ \quad \cdot \delta_{IX} \cdot p^X(w[i, l-1], T[j]), \\ p_O^I(\sigma(w[i], \sigma, w[k]), \lambda) \\ \quad \cdot \delta_{IX} \cdot p^X(w[i+1, k-1], T[j]), \\ p_O^I(\sigma(w[i], \sigma), \lambda) \\ \quad \cdot \delta_{IX} \cdot p^X(w[i+1, k], T[j]), \\ p_O^I(\sigma(\sigma, w[k]), \lambda) \\ \quad \cdot \delta_{IX} \cdot p^X(w[i, k-1], T[j]). \end{array} \right.$$

$$p^D(w[i, k], T[j]) = p_O^D(\epsilon, v(j)) \cdot \max_{X \in \{D, M\}} \left\{ \begin{array}{l} p^D(\epsilon, T[j_1]) \cdot \delta_{DX} \cdot p^X(w[i, k], T[j_2]), \\ \quad \text{if a node } j \text{ is of arity 2,} \\ p^D(\epsilon, T[j_2]) \cdot \delta_{DX} \cdot p^X(w[i, k], T[j_1]), \\ \quad \text{if a node } j \text{ is of arity 2,} \\ \delta_{DX} \cdot p^X(w[i, k], T[j_1]), \\ \quad \text{if a node } j \text{ is of arity 1.} \end{array} \right.$$

$$p^X(\epsilon, \theta) = 1, \quad \text{for } X \in \{M, I, D\},$$

where we consider $\sigma(\sigma, \sigma)$ as a node of arity 2 to represent branched structures, $\sigma(w[i], \sigma, w[k])$ as a node of arity 1 labeled with a pair of symbols $(w[i], w[k])$, and $\sigma(w[i], \sigma)$, $\sigma(\sigma, w[k])$ as nodes of arity 1 labeled with a symbol $w[i]$, $w[k]$, respectively.

An optimal structural alignment between a sequence w and a skeletal tree T is obtained by calculating $\max\{\tau_M \cdot p^M(w[1, n], T[1]), \tau_I \cdot p^I(w[1, n], T[1]), \tau_D \cdot p^D(w[1, n], T[1])\}$ for some predefined initial probabilities τ_M, τ_I, τ_D .

A similar approach was taken for the structural alignment problem of RNAs by Jiang *et al.* (2002). Our advantage of PHMMTSs is that various types of structural alignments such as affine-gap alignments are available in the automata-theoretic framework.

A new type of gap penalty

We more closely look into the recurrence equation at the deletion state for the structural alignment, that is, $p^D(w[i, k], T[j])$. At the branching node labeled $\sigma(\sigma, \sigma)$ of arity 2, we skip a whole subtree or equivalently a whole substructure of RNA and assign a probability by calculating $p_O^D(\lambda, v(j)) \cdot p^D(\epsilon, T[j_1]) \cdot \delta_{DX} \cdot p^X(w[i, k], T[j_2])$. In this formula, the probability $p^D(\epsilon, T[j_1])$ corresponds to a probability of skipping the whole subtree (substructure) $T[j_1]$. In our actual experiments, we set $p^D(\epsilon, T[j_1])$ at a constant probability independently of $T[j_1]$, which implies that a constant gap penalty is given to a whole substructure to be skipped independently of its size (length). This proposes a new type of gap penalty which is suitable for skipping a whole substructure independently of its length and avoids an exponentially decaying distribution. This gap penalty may relate to *generalized HMMs* (Pachter *et al.*, 2002) and be applicable to aligning a full-length cDNA to genomic sequence.

Pair stochastic context-free grammars

Stochastic context-free grammars (SCFGs) are a more powerful class of stochastic grammars than the HMMs in the Chomsky hierarchy of formal languages, and have been successfully applied to the problems of folding, aligning and modeling families of homologous RNA sequences (Sakakibara *et al.*, 1994; Eddy and Durbin, 1994). Pair stochastic context-free grammars (PSCFGs) are a generalization of stochastic context-free grammars and can generate an aligned pair of sequences. PSCFGs have been studied for alignments of a pair of RNA sequences without any prior information about their secondary structures (Rivas and Eddy, 2001; Holmes and Rubin, 2002).

On the other hand, it is known that the set of derivation (parse) trees of a CFG, and further the set of skeletal trees of a CFG can be recognized by a tree automaton. Combined with these observations, PHMMTSs with input of a pair of two linear sequences imply the PSCFGs. A serious problem using PSCFGs is its expensive computational cost to execute a parsing process of PSCFG.

Match state:

	(G, C)	(C, G)	(A, U)	(U, A)	(G, U)	(U, G)
(G, C)	4	3	3	3	3	3
	(G, G)	(G, A)	(A, C)	(C, C)	(U, C)	(A, A)
(G, C)	-2	-2	-2	-2	-2	-6
	(C, A)	(A, G)	(C, U)	(U, U)		
(G, C)	-6	-6	-6	-6		
	G	C	A	U		$\sigma(\sigma, \sigma)$
G	1	-3	-3	-3	$\sigma(\sigma, \sigma)$	5

Insertion state:

Equal emission probabilities 0.25 for pairs of labels of the form (X, λ) for $X \in \{G, C, A, U\}$, and 0 emission probabilities for all other pairs.

Deletion state:

Equal probabilities for all pairs of labels of the form (ϵ, V) , where V is any label.

Fig. 5. An example of score matrix and emission probabilities on three states.

EXPERIMENTAL RESULTS

We have designed a dynamic programming algorithm to calculate the recurrence equations for the structural alignment and implemented a prototype system to execute the algorithm. We have done two experiments on RNA families of Transfer RNAs and Hammerhead ribozyme to test the structural alignment algorithm. In our experiments, we use the emission probabilities based on the score matrix shown in Figure 5, a part of which is employed from Gorodkin *et al.* (1997).

Transfer RNA

The trusted sequences and alignments of tRNA sequences with the annotations of secondary structures were taken from the database of Steinberg *et al.* (1993), and are organized into seven groups and denoted by their databank codes: 1. ARCHAE (*archaebacteria*), 2. CY (*cytoplasm*), 3. CYACHL (*cyanelle and chloroplast*), 4. EUBACT (*eubacteria*), 5. VIRUS (*viruses*), 6. MT (*mitochondria*) and 7. PART III (Part III). We obtain 1222 unique sequences, each between 51 and 93 bases long, by omitting duplicate primary sequences and sequences containing unusual characters. The number of sequences contained in each group is: 103 in ARCHAE, 230 in CY, 184 in CYACHL, 201 in EUBACT, 24 in VIRUS, 422 in MT, and 58 in PART III.

In our experiment, we have randomly chosen one tRNA sequence annotated with the known secondary structure from the group EUBACT in the database, which is shown in Figure 6, and structurally aligned all other ‘unfolded’

Input:

```

((((((( ((((      )))) ((((((      )))))
GGGCCUUAGCUCAGCUGGGAGAGACCCUGCCUUGCACGCAGGG

      ((((((      )))))))))))
GGUCGACGGUUCGAUCCCGUAGGGUCCA
    
```

Fig. 6. A target tRNA sequence with secondary structure to be structurally aligned into.

Prediction results of secondary structures for unfolded tRNA sequences:

tRNA type	ARCHAE	CY	CYACHL	EUBACT
PHMMTS	95.47%	99.71%	97.59%	98.65%
Clustal-W	90.06%	94.29%	98.13%	94.12%

tRNA type	VIRUS	MT	PART III
PHMMTS	100.00%	85.38%	50.22%
Clustal-W	65.08%	57.78%	13.64%

Fig. 7. Fraction of correct base-pairs predicted by the structural alignment algorithm based on PHMMTSs (upper), and predictions based on multiple alignment by Clustal-W (lower).

tRNA sequences into the ‘folded’ tRNA sequence. We counted the fraction of base pairs specified by the trusted alignment that matched in the predictions of our structural alignment algorithm. We also compared the prediction results of our structural alignment algorithm with the predictions based on multiple alignments by a standard alignment software ‘Clustal-W’ (Thompson *et al.*, 1994). Clustal-W was input a set of unfolded tRNA sequences in each group together with the target tRNA sequence shown in Figure 6, and secondary structures for each set of tRNA sequences were predicted based on a multiple alignment made by Clustal-W. Clustal-W does not take into account the secondary structure of the target tRNA sequence when it makes alignments, but has an advantage of taking multiple alignment instead of pairwise alignment. The results are shown in Figure 7. The predicted secondary structures of our structural alignment algorithm agree extremely well (almost 100%) with the trusted alignment, and it outperforms Clustal-W, especially for the groups of VIRUS, MT, and PART III. The group PART III is most difficult in the sense that it contains 58 tRNA sequences of unusual secondary structures lacking a whole loop of the D-domain, and for the group, our prediction accuracy is still 50% while Clustal-W predicts very poorly. From these comparisons, it is very clear that the structural alignment is essential to align RNA sequences and predict the secondary structures.

Prediction results:

	(trained CM)	PHMMTS	Clustal-W
Hammerhead ribozyme	(100.00%)	81.89%	46.29%

An example of structural alignment:

```

[Target RNA sequence with structure
annotation]
      (((((((((( ))) )))) ))))((( )))
GACUGAUGAGUCCGAAAGGACGAAACACCAGUG
* ***** ** ** ***** **
GGCUGAUGAGU-CGUGAG-ACGAAACAC-CUUG
      (((((( ( ( ) ) )))) ))))((( )))
[Predicted alignment of an unfolded sequence]

* ***** ***** **
GGCUGAUGAGUCGU..GAGACGAAACAC.CUUG
      (((((((((( ))) )))) ))))((( )))
[Trusted alignment in Rfam database
for the unfolded sequence]
    
```

Fig. 8. Fraction of correct base-pairs predicted by our structural alignment algorithm with a randomly chosen sequence with secondary structure annotation for Hammerhead ribozyme (upper) and an example of structural alignment (lower).

Hammerhead ribozyme

A similar experiment was done for the RNA family of Hammerhead ribozyme. The dataset was taken from the RNA families database ‘Rfam’ at Sanger Institute (Griffiths-Jones *et al.*, 2003). Those RNA sequences are aligned and annotated with secondary structures by using the Covariance Model (CM) method (Eddy and Durbin, 1994), which is a SCFG-based method for modeling RNA sequences. The results shown in Figure 8 represent in fact a performance comparison between our structural alignment algorithm and a well-trained CM method. One important fact is that the Covariance Model used for the alignments and annotations was well trained on a number of RNA sequences and specialized to the hammerhead ribozyme RNA family, while our structural alignment algorithm does not require such training process and predicts secondary structures only based on structural alignment to one single folded RNA sequence.

RELATED WORKS AND DISCUSSIONS

Related works

There are several related works to our approach. Most related are our previous work of the stochastic context-free grammar (SCFG) (Sakakibara *et al.*, 1994) and the covariance model (CM) (Eddy and Durbin, 1994) for modeling RNA sequences. One significant difference

from our PHMMTS-based method is that those methods require complicated steps of designing an initial grammar, training parameters for the grammar and enough sequences for the training specific to a certain family of RNAs. Our method does not require such steps, and it is just input a pair of one target folded RNA sequence and one unfolded RNA sequence, and general enough to be applied to any family of RNAs.

Pair SCFGs (Rivas and Eddy, 2001; Holmes and Rubin, 2002) and related works (Gorodkin *et al.*, 1997; Mathews and Turner, 2002) such as Sankoff's algorithm (Sankoff, 1985) are more challenging to align two unfolded RNA sequences of unknown structure. However, those methods are computationally expensive, and the methods may avoid such impractical computation time by taking pre-aligned sequences as input. Aligning two known secondary structures of RNA sequences (Shapiro and Zhang, 1990; Jiang *et al.*, 2002) is also related to our work and is clearly reduced to the problem of alignment of trees. As we mentioned, our PHMMTSs provide a unifying framework for any studies to compare two RNA sequences such as alignments of trees, structural alignments and PSCFGs.

Computational complexity

The computational cost to execute PHMMTSs for structural alignments is theoretically of the same order as the computational complexity to parse an input sequence with SCFGs. More precisely, the computational complexity to run a PHMMTS for an input pair of an unfolded sequence of length N and a skeletal tree of size M is $O(KMN^3)$, where K is the number of states in the PHMMTS. The computational complexity to parse an input sequence of length N with a SCFG of size L is $O(LN^3)$, and in general, the size L of a profile SCFG specific to a family of RNAs is proportional to K and the size M of a skeletal tree for a representative folded RNA sequence in the family of RNAs. On the other hand, the computational complexity to train a SCFG of size L is $O(JL^3N^3)$ by using some parameter-estimation algorithms (Lari and Young, 1990; Sakakibara *et al.*, 1994) for SCFGs, where J is the number of training sequences and N is the maximum length of training sequences.

The executions of pair SCFGs usually require $O(LM^3N^3)$ computation time for a pair of input sequences of length M and N by using dynamic programming, where L is the size of the PSCFG.

Future work

Our approach has a couple of advantages compared with existing methods. Our structural alignment method is fundamentally a pairwise alignment based on dynamic programming so that many techniques (Durbin *et al.*, 1998) developed for alignments of sequences can be

applied. For example, the Smith–Waterman algorithm can be applied to obtain *local alignment* for database search, and the Myers–Miller algorithm can also be applied to reduce the space complexity.

There are several remaining problems to make our method practically usable. Most important is to refine emission probabilities and related score matrix by calculating from existing reliable alignments for RNA families.

ACKNOWLEDGEMENTS

This work was performed in part through Special Coordination Funds for Promoting Science and Technology from the Ministry of Education, Culture, Sports, Science and Technology, the Japanese Government.

REFERENCES

- Aho, A. and Ullman, J. (1972) *The Theory of Parsing, Translation and Compiling*, Vol. I: Parsing, Prentice Hall, Englewood Cliffs, NJ.
- Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. (1998) *Biological Sequence Analysis*. Cambridge University Press, Cambridge.
- Eddy, S. and Durbin, R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res.*, **22**, 2079–2088.
- Eddy, S. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
- Eddy, S. (2001) Noncoding RNA genes and the modern RNA world. *Nat. Rev. Genet.*, **21**, 919–929.
- Griffiths-Jones, B., Bateman, A., Marshall, M., Khanna, A. and Eddy, S. (2003) Rfam: an RNA family database. *Nucleic Acids Res.*, **31**, 439–441. <http://www.sanger.ac.uk/Software/Rfam/>.
- Gorodkin, J., Heyer, L. and Stormo, G. (1997) Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, **25**, 3724–3732.
- Holmes, I. and Rubin, G. (2002) Pairwise RNA structure comparison with stochastic context-free grammars. In *Proceedings of 5th Pacific Symposium on Biocomputing*. World Scientific Press, Singapore, pp. 163–174.
- Jiang, T., Wang, L. and Zhang, K. (1995) Alignment of trees—an alternative to tree edit. *Theor. Comp. Sci.*, **143**, 137–148.
- Jiang, T., Lin, G., Ma, B. and Zhang, K. (2002) A general edit distance between RNA structures. *J. Comput. Biol.*, **9**, 371–388.
- Knudsen, B. and Hein, J. (1999) RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, **15**, 446–454.
- Lari, K. and Young, S. (1990) The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, **4**, 35–56.
- Mathews, D. and Turner, D. (2002) Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.*, **317**, 191–203.
- Pachter, L., Alexandersson, M. and Cawley, S. (2002) Applications of generalized pair hidden Markov models to alignment and gene finding problems. *J. Comput. Biol.*, **9**, 389–399.
- Rivas, E. and Eddy, S. (2001) Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, **2:8** (10 October 2001).

- Sakakibara, Y., Brown, M., Hughey, R., Mian, I.S., Sjolander, K., Underwood, R.C. and Haussler, D. (1994) Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Res.*, **22**, 5112–5120.
- Sakakibara, Y. (1992) Efficient learning of context-free grammars from positive structural examples. *Information and Computation*, **97**, 23–60.
- Sankoff, D. (1985) Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, **45**, 810–825.
- Schattner, P. (2002) Searching for RNA genes using base-composition statistics. *Nucleic Acids Res.*, **30**, 2076–2082.
- Searls, D. and Murphy, K. (1995) Automata-theoretic models of mutation and alignment. *Proceedings of 3rd International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 341–349.
- Shapiro, B. and Zhang, K. (1990) Comparing multiple RNA secondary structures using tree comparisons. *Comput. Appl. Biosci.*, **6**, 309–318.
- Steinberg, S., Misch, A. and Sprinzl, M. (1993) Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res.*, **21**, 3011–3015.
- Thompson, J., Higgins, D. and Gibson, T. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.