Original Paper

Stochastic Regular Approximation of Tree Grammars and Its Application to Faster ncRNA Family Annotation

Kazuya Ogasawara† and Satoshi Kobayashi†

Tree Adjoining Grammar (TAG) is a useful grammatical tool to model RNA secondary structures containing pseudoknots, but its time complexity for parsing is not small enough for the practical use. Recently, Weinberg and Ruzzo proposed a method of approximating stochastic context free grammar by stochastic regular grammar and applied it to faster genome annotation of non-coding RNA families. This paper proposes a method for extending their idea to stochastic approximation of TAGs by regular grammars. We will also report some preliminary experimental results on how well we can filter out non candidate parts of genome sequences by using obtained approximate regular grammars.

1. Introduction

Biological sequences contain both of stochastic and structural information. Formal grammars are quite useful tools for modeling, with high accuracy, such stochastic and structural features of biological sequences. Covariance Model [7),24)], CM for short, is one of the most successful grammatical model for RNA families, in which stochastic context free grammar is used to model RNA primary and secondary structures. This seminal grammatical technique makes it possible to produce reliable RNA databases [26)].

A secondary structure of an RNA sequence $w = a_1 \cdots a_n$ ($a_i \in \{A, C, G, U\}$) is base pairing information between bases in w, which is described as a finite set of integer pairs (i, j) with i and j ($1 \leq i < j \leq n$) indicating i-th and j-th bases in w, respectively. It is recognized as an important issue to deal with secondary structure of a given RNA sequence, since its structure has strong relation to its biological function. However, one of the difficulties in the CM approach is that it can not model the secondary structure, called pseudoknot, which commonly appear in various RNA molecules and play various roles in biological functions [5),16)].

When an RNA secondary structure contains base pairs (i, j) and (i', j') such that $i < i' < j < j'$ or $i' < i < j' < j$, it is called a *pseudoknot* (See **Fig. 1**). Since the crossing dependency in a pseudoknot can not be represented by context free grammars, in order to deal with pseudonot, it is necessary to prepare a grammatical model whose generative capacity is beyond context freeness. There are several candidate grammars proposed, including Tree Adjoining Grammar (TAG) [13),30),31)], RNA Pseudoknot Grammar [22)], Multiple Context Free Grammar [11),12)], CFG based Parallel Communicating Grammar [3)], etc.

A Tree Adjoining Grammar (TAG) is a grammatical device to generate a set of trees rather than a set of strings, which was first proposed by Joshi and Takahashi [10)]. It is known that the string languages generated by TAGs are between context sensitiveness and context freeness, and can model pseudoknotted structures.

Although TAG has enough computational capacity to precisely analyze pseudoknots, its time complexity for parsing is not small enough for the practical use. Recently, Weinberg and Ruzzo proposed a method of approximating stochastic context free grammar by stochastic regular grammar and applied it to faster genome annotation of non-coding RNA families using CM method [32)]. The idea is to use such an approximate stochastic regular grammar to filter total genome sequences and find out candidate positions of the target family. CM is applied only to such candidates, which drastically reduces the total time of annotation of ncRNA families. This paper extends their idea and applied it to approximating stochastic tree grammars by regular grammars. We will also report some preliminary experimental results on how well the obtained approximate regular grammars can be used to filter out non

† Department of Computer Science, the University of Electro-Communications

Preliminary version of this paper appeared in Proc. of 1st international Conference on Language and Automata Theory and Applications (LATA'2007).

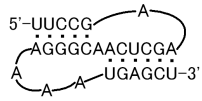

Fig. 1 The graphical representation of pseudoknot.

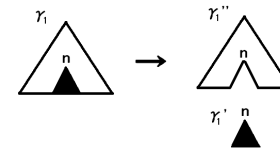

Fig. 2 Split γ_1 at n into $\gamma_1^{'}$ and $\gamma_1^{''}$.

candidate parts of genome sequences.

The purpose of this paper is to show the effectiveness of the proposed approximation method to filter out non candidate positions of ncRNA families in total genome sequences. The computational experiment of the ncRNA family annotation by TAG still requires much computation time. But, the authors believe that the technique used in this paper can also be extendedly used in the parsing process of TAG itself. Thus, the results reported in this paper is a challenging first step toward the goal where we can *efficiently* and *effectively* use TAG for the genome annotation with high accuracy.

In Section 2, we will give definitions and notations of tree adjoining grammars and describe how we can model RNA secondary structures including pseudoknots using TAGs. Inspired from the results by Weinberg and Ruzzo, in Section 3, we will propose a method of approximating a given stochastic tree adjoining grammar for RNA modeling to a stochastic regular grammar. We will also explain a method of tuning stochastic parameters. In Section 4, some preliminary experimental results are reported, in which we will show the effectiveness of the proposed method. In Section 5, we will show related works. Finally, Section 6 gives some concluding remarks.

2. Tree Adjoining Grammar for RNA Secondary Structure

2.1 Tree Adjoining Grammar

A TAG is a grammatical device for generating trees. Let us consider a tree labeled with symbols in the alphabet $V = N \cup \Sigma$ where N and Σ are disjoint. Symbols in N are called *nonterminals*, and those in Σ are called *terminals*. By τ_V, we denote the set of trees whose internal and leaf nodes are labeled by symbols in N and V, respectively. A TAG G is defined by $G = (V, C, A)$, where V is an alphabet, and C and A are finite subsets of τ_V such that every $t \in C$ does not have a leaf node with a label in N, every $t \in A$ has exactly one leaf node with a label in N. Elements of C and A are called *center trees* and *adjunct trees*, respectively. Elements of $C \cup A$ are called *elementary trees*. By definition, the leaf node of an adjunct tree whose label is nonterminal is determined uniquely and is called a *foot node*. Furthermore, an additional requirement for an adjunct tree is that the label of the foot node should be equivalent to that of the root node. The path from the root to the foot node of an adjunct tree is called a *spine*. It is often the case that some constraints are associated with each node n of elementary trees. In this paper, we will consider the following constraints:

(1) Null Adjoining (NA): No adjunct trees can be adjoined at n.
(2) Obligating Adjoining (OA): a member of the set A must be adjoined at n.

A node without NA constraint is said to be *active*. These constraints play an important role in defining the derivation process of tree grammars.

Let $G = (V, C, A)$ be a TAG. An adjunct tree α is adjoinable to a tree $\gamma_1 \in \tau_V$, if γ_1 has an active internal node n whose label is the same as that of the root node of α. The operation to *adjoin α on n of γ_1* is defined by the following procedure:

Split: Split γ_1 at n into two trees so that n is contained duplicatedly in both of the trees. Let $\gamma_1^{'}$ be the tree consisting of the nodes (including n) which are located below the node n in γ_1. Let $\gamma_1^{''}$ be the tree consisting of the nodes (including n) which are not located below n. The tree $\gamma_1^{'}$ is called a subtree of γ_1 at n, and $\gamma_1^{''}$ is called a supertree of γ_1 at n (See **Fig. 2**).

Merge: Combine $\gamma_1^{''}$ and α by identifying n of $\gamma_1^{''}$ and the root node of α, and then combine again the resultant tree and $\gamma_1^{'}$ by identifying the foot node of α and the root node (i.e. n) of $\gamma_1^{'}$ (See **Fig. 3**).

We write $\gamma_1 \vdash_G \gamma_2$ if there is an adjunct tree $\alpha \in A$ which is adjoinable to γ_1 and γ_2 is obtained by adjoining α to γ_1. By $\vdash_G^*$, we denote the reflexive and transitive closure of $\vdash_G$. We define:

$$\tau(G) = \{t \in \tau_V \mid t_0 \vdash_G^* t, t \text{ does not contain a node with OA constraint and } t_0 \in C.\ \}$$

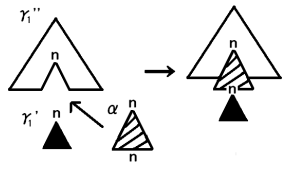

Fig. 3 Merge $\gamma_1^{'}$, α, and $\gamma_1^{''}$.

$L(G) = \{Y(t) \in \Sigma^* \mid t \in \tau(G)\}$,
where $Y(t)$ is the yield of a tree t, which is defined as the string consisting of labels of leaf nodes of t ordered in depth-first and left-to-right.

2.2 TAG_{RNA}

In this paper, we will focus on a subclass of TAGs, called TAG_{RNA}, which was proposed by one of the authors in order to model RNA secondary structures including pseudoknots [13),30),31)]. Although the generative capacity of TAG_{RNA} is strictly weaker than that of TAGs ☆, it can model various types of existing RNA secondary structures.

A center tree of a TAG_{RNA} should be of the form represented in **Fig. 4** (a), where $S \in N$ and λ is an empty string. An active node is indicated by the symbol '*'. A center tree used in a TAG_{RNA} always has only two nodes root and leaf. The root node is an active node and the label of the leaf node is always λ. An adjunct tree of a TAG_{RNA} should be of the form represented in Fig. 4 (b) or Fig. 4 (c), where $e_1, e_2, e_3, e_4 \in \Sigma \cup \{\circ\}$ and $X, Y, Z \in N$. If $e_{i\ (i=1,2,3,4)}$ is '$\circ$', e_i is said to be a blank node (the blank node indicates that the node doesn't exist.). We call an adjunct tree of Fig. 4 (b) a *type-A* adjunct tree which has the root node, the foot node, one active node on the spine, and four nodes (leaf or blank). In a type-A adjunct tree, the root node has three child nodes. These are located down left (the upper-left node of the active node), just under (the active node on the spine), and down right (the upper-right node of the active node) of the root node respectively. The active node also has three child nodes. These are located down left, just under (the foot node), and down right of the active node, respectively. We call an adjunct tree of Fig. 4 (c) a *type-B* adjunct tree which has the root node, the foot node, and two active nodes. One of two active nodes is on the spine, and the other is located at either upper-left, upper-right, down-left or down-right of the active node on the spine. The latter active node has a leaf node as a child, whose label is always λ. In this paper, the latter active node is called a *branch active node*.

☆ In TAG_{RNA}, an adjunct tree has at most one active node on its spine, which restricts the generative capacity of TAG_{RNA}, as tree and string languages, compared to original TAG.

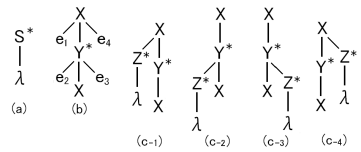

Fig. 4 Elemental trees of TAG_{RNA}.

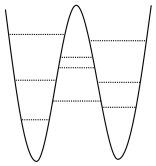

Fig. 5 The graphical representation of a W-shaped structure.

For convenience of the discussion in the sequel, we introduce notations of elementary trees of TAG_{RNA} as follows:
By $[S^* - \lambda]$, we denote a center tree of TAG_{RNA} whose root label is $S \in N$. By $[X \rightarrow Y^*(e_1, e_2, e_3, e_4)]$, we denote a type-A adjunct tree of TAG_{RNA} such that root label is $X \in N$, active node label is $Y \in N$, the upper-left node label is e_1, the down-left is e_2, the down-right is e_3, and the upper-right is e_4, respectively $(e_1, e_2, e_3, e_4 \in \Sigma \cup \{\circ\})$. By $[X \rightarrow Y^*(Z^*, \circ, \circ, \circ)]$, we denote a type-B adjunct tree of TAG_{RNA}, where $X \in N$ is the label of the root, $Y \in N$ is the label of the active node on the spine, and $Z \in N$ is the label of the branch active node. In this case, the upper-left node is active, and down-left, down-right, upper-right nodes are the blank node (See Fig. 4 (c-1)). Notations $[X \rightarrow Y^*(\circ, Z^*, \circ, \circ)]$, $[X \rightarrow Y^*(\circ, \circ, Z^*, \circ)]$, and $[X \rightarrow Y^*(\circ, \circ, \circ, Z^*)]$ are introduced in a similar way (See Fig. 4 (c-2), Fig. 4 (c-3), Fig. 4 (c-4), respectively).

A TAG_{RNA} can be used to model an RNA secondary structure including pseudoknots. Asakawa characterized the class of RNA secondary structures represented by TAG_{RNA}. He proved that TAG_{RNA} pseudoknotted structure can be characterized by a W-shaped structure (see **Fig. 5**), where base pairs are represented by dotted horizontal lines. Almost all of the pseudoknot structures in Rfam can be modeled by a TAG_{RNA} with some exceptions.

For instance, let us consider an RNA sub-

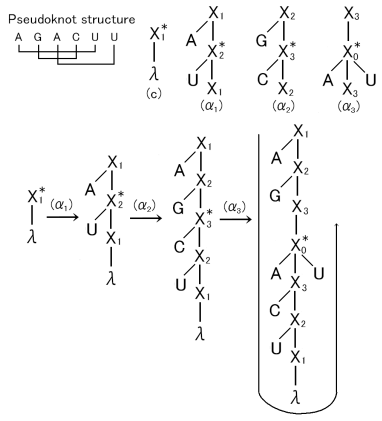

Fig. 6 The derivation process of w by TAG_{RNA}.

sequence of $w = AGACUU$ with a secondary structure $\{(1,5),(2,4),(3,6)\}$. The set of elementary trees to model this RNA is given by (c) $[X_1^* - \lambda]$, (α_1) $[X_1 \rightarrow X_2^*(A,U,\circ,\circ)]$, (α_2) $[X_2 \rightarrow X_3^*(G,C,\circ,\circ)]$ and (α_3) $[X_3 \rightarrow X_0^*(\circ,A,U,\circ)]$. Its derivation process is shown in **Fig. 6**.

2.3 Automatic Construction of Stochastic TAG

Takakura, et al., developed a system for generating stochastic TAG from multiple alignment data of RNA sequences with secondary structure information [28),29)]. This system is based on the algorithm developed by Asakawa which decides whether or not a given secondary structure can be modeled by a TAG [2)]. By modifying this algorithm and collecting stochastic information of primary sequences in the alignment data, Takakura developed an efficient system for generating stochastic TAG which models the given alignment data. This system can generate stochastic TAG in $O(n^2m)$, where n is the maximum length of RNA sequence and m is the number of RNA sequences in the given alignment data ☆.

Although we have not yet developed a learning algorithm of stochastic TAG like that of stochastic CFG in CM method, the simple implementation of collecting stochastic information from the alignment data works fairy well for the unknown data classification [28)]. We will use Takakura's system for the experiments in Section 4.

☆ Theoretically, we can develop an algorithm which runs in time $O(nm)$, but the current implementation is $O(n^2m)$.

3. Approximating Tree Grammars by Regular Grammars

3.1 Speed-up by Filtering

Although TAG_{RNA}'s can effectively model RNA secondary structures including pseudoknots, its time complexity for parsing is $O(n^5)$ (n is the length of input string) [31)], which makes it hard to apply TAG_{RNA}'s to the search of functional RNAs in large genome databases. In order to overcome the difficulty, we will propose a method for making a TAG_{RNA} search efficient without loss of its accuracy based on Weinberg's idea. The method constructs, from a stochastic TAG_{RNA}, a stochastic regular grammar (SREG), which approximates the given TAG_{RNA} and filters genome databases. Stochastic parsing by TAG_{RNA} is applied only to candidates of functional RNAs which pass through the filtering process by the SREG. Since SREG parsing is much faster than TAG_{RNA} parsing, the good approximation of TAG_{RNA} by SREG makes the database search much faster, even if we consider the time of approximating TAG_{RNA}.

Let $P(w \mid G)$ be the maximum probability of all derivations of G which generate w. An approximate SREG G_{reg} for a TAG_{RNA} G_{tag} should satisfy:

$$P(w \mid G_{reg}) \geq P(w \mid G_{tag}), \quad (\text{for any } w) \tag{1}$$

in order to guarantee that we do not loose any candidate functional RNAs at the filtration stage.

We will describe the construction of an approximate SREG for a TAG_{RNA} satisfying the above constraint in two steps:

(1) Construct a regular grammar from a stochastic TAG_{RNA} without considering probabilistic parameters.
(2) Determine the probability of each production rule of an SREG.

3.2 Regular Approximation of a Stochastic TAG_{RNA}

The approximation method proposed by Weinberg and Ruzzo [32)] is essentially the same as that by Nederhof [17)]. Although it is possible to apply CFG approximation methods by Harbusch or Poller [8),19)] to a given TAG and then apply Nederhof's regular approximation to it, we will extendedly apply regular approximation technique similar to Nederhof's [17)] directly to TAGs for its simplicity.

We introduce some technical terms before showing the method of constructing an approximate SREG for a given TAG_{RNA}. A node with nonterminal symbol Z is said to be an *initial node* if there exists no $\gamma \in A$ such that $\gamma = [X \to Z^*(a,b,c,d)]$ for some $a,b,c,d \in \Sigma \cup \{\circ\}$ and some $X \in N$ with $X \neq Z$.

Let β be an adjunct tree. By UL(β) (DL(β), DR(β), UR(β), respectively), we denote the substring of Y(β) which is located at the upper-left (UL) (down-left (DL), down-right (DR), upper-right (UR), respectively) segment from the active node in β.

In order to construct a regular approximation of a TAG_{RNA} we will convert an adjunct tree β into four regular production rules each corresponding to one of segments UL(β), DL(β), DR(β), UR(β). For instance, consider a type-A adjunct tree β denoted by $[X \to Y^*(a,b,c,d)]$ $(X,Y \in N,\ a,b,c,d \in \Sigma \cup \{\circ\})$. Production rules corresponding to β are as follows:

$$X^{UL} \to aY^{UL},\ Y^{DL} \to bX^{DL},$$
$$X^{DR} \to cY^{DR},\ Y^{UR} \to dX^{UR}. \qquad (2)$$

However, these production rules are independent and not connected to each other. We need to construct special production rules which connect these production rules. If the node denoted by X is an initial node, we construct a special production rule which connects DL and DR segments: $X^{DL} \to X^{DR}$. In this paper, we call this special production rule *a connecting production rule*.

From the behavior of the adjoin operation, we know that DL and DR segments of an initially applied adjunct tree are located consecutively in the yield of a resultant tree. Thus, we only have to construct a special production rule which connects DL and DR segments according to the label of an initial node.

Second, we show how to construct production rules in the case of a type B adjunct tree. Let β be a type B adjunct tree denoted by $[X \to Y^*(Z^*,\circ,\circ,\circ)]$ $(X,Y,Z \in N)$. Production rules corresponding to β are as follows:

$$X^{UL} \to Z^{UL},\ Z^{UR} \to Y^{UL},$$
$$Y^{DL} \to X^{DL},\ X^{DR} \to Y^{DR},$$
$$Y^{UR} \to X^{UR}. \qquad (3)$$

Because the branch active node (in this case, the node denoted by Z) is always an initial node, we also construct a connecting production rule: $Z^{DL} \to Z^{DR}$.

In cases of $[X \to Y^*(\circ,Z^*,\circ,\circ)]$, $[X \to Y^*(\circ,\circ,Z^*,\circ)]$, $[X \to Y^*(\circ,\circ,\circ,Z^*)]$, production rules are introduced in a similar way.

Third, we show how to construct special production rules which connect UL and DL segments, DR and UR segments respectively. UL and DL segments, and DR and UR segments are separated at the active node which was introduced at the final step of the derivation. Thus, we only have to construct special production rules according to the label of such a terminating node X:

$$X^{UL} \to X^{DL},\ X^{DR} \to X^{UR}.$$

Also, we call these special production rules connecting production rules.

3.3 Constraints on Stochastic Parameters

In our RNA secondary structure analysis system, a 0th-order Hidden Markov model is used as a random model to be compared with a stochastic TAG_{RNA} model, in order to judge whether a given RNA sequence can be a member of the family. Let $P(w \mid G_{rand})$ be the probability of an RNA subsequence w generated by a random model G_{rand}. Let $P(w \mid G_{tag})$ be the probability of w generated by a TAG_{RNA} G_{tag}. We use a random model as threshold, i.e., w belongs to an RNA family modeled by G_{tag}, if the following inequality Eq. (4) holds:

$$P(w \mid G_{tag}) \geq P(w \mid G_{rand}). \qquad (4)$$

In order to construct a stochastic regular grammar G_{reg} which approximates G_{tag}, it is important to assign probabilities so that Eq. (1) holds, since $P(w \mid G_{rand}) > P(w \mid G_{reg})$ and Eq. (1) imply $P(w|G_{rand}) > P(w|G_{tag})$, which means that the inequality $P(w|G_{rand}) > P(w|G_{reg})$ can be used at the filtration stage.

In this subsection, we use the logarithm of the probability instead of the probability itself. We will show how to construct probabilities of production rules satisfying Eq. (1). Consider a type-A adjunct tree β denoted by $[X \to Y^*(a,b,c,d)]$ and let L_1 be the logarithm of the probability of β. According to our algorithm, we obtain the production rules shown in Eq. (2) from β. Let ℓ_1, ℓ_2, ℓ_3 and ℓ_4 be the logarithm of probabilities of these production rules, respectively. In order to satisfy Eq. (1), it suffices to use ℓ_1, ℓ_2, ℓ_3 and ℓ_4 satisfying the following inequality constraint:

$$\ell_1 + \ell_2 + \ell_3 + \ell_4 \geq L_1.$$

The probability of connecting production rules is always assigned to be 1.

Next, we consider a type-B adjunct tree γ denoted by $[X \to Y^*(Z^*,\circ,\circ,\circ)]$. Let L_2 be the

logarithm of the probability of γ. We obtain the production rules shown in Eq. (3) from γ. In a similar manner as in the case of type-A adjunct tree, the problem is reduced to the one to find ℓ_5, ℓ_6, ℓ_7, ℓ_8 and ℓ_9 satisfying:

$$\ell_5 + \ell_6 + \ell_7 + \ell_8 + \ell_9 \geq L_2,$$

where ℓ_5, ℓ_6, ℓ_7, ℓ_8 and ℓ_9 correspond to the logarithm of the probabilities of $X^{UL} \rightarrow Z^{UL}, \ldots, Y^{UR} \rightarrow X^{UR}$, respectively. We construct a set of inequality constraints for all adjunct trees and denote it by CN.

3.4 Improving Stochastic Parameters

In this subsection, we will show a method to improve the filtering efficiency. The probabilistic parameters satisfying the set CN of constraints always meet the filtration condition (Eq. (1)). However, we aim to remove candidates that can not be a member of the family modeled by G_{tag} as many as possible. Following the works by Weinberg and Ruzzo, we will propose a method for tuning probabilistic parameters based on nonlinear programming to meet such a requirement.

Parameter tuning method proposed in this subsection uses two grammar models, G_{tag} and G_{rand} bellow, in order to improve the approximation accuracy of an SREG obtained from G_{tag} as in the previous subsection. Thus, it is different from Kullback-Leibler distance minimization approach [18].

For $w \in L(G_{reg})$, let $Pr(w \mid G_{reg})$ be the probability of w generated by G_{reg}. Let π be a derivation *path* to generate w by G_{reg}. Let $Pr(w \mid \pi)$ be the probability of generating w based on the path π. Let Π_w be the set of all paths for generating w. Then, we have:

$$Pr(w \mid G_{reg}) = \sum_{\pi \in \Pi_w} Pr(w \mid \pi).$$

We use G_{rand} as a random model for whole genome sequences (0th order HMM based on a, g, c, u frequencies). We define the objective function OB as follows:

$$OB = \sum_{w \in L(G_{reg})} Pr(w|G_{reg}) Pr(w|G_{rand}).$$

By minimizing OB subject to the set CN of constraints, the number of candidate RNAs might be reduced.

The structure of the grammar G_{reg} can be explicitly represented by converting it to a corresponding automaton M such that $L(G_{reg}) = L(M)$, where each production rule $S_i \rightarrow xS_j$ is transformed into a transition from $S_i \rightarrow S_j$ with a label x. The approximate regular grammar G_{reg} has a simple linear structure if we neglect self-loops of the form $S_i \rightarrow xS_i$

For easy computation of OB, we will restrict Π_w to the set of paths π_w such that each self-loop is contained at most once in π_w.

Then, we can efficiently compute OB with a dynamic proagramming method, as follows:

$$OB(S_i) = OB(S_{i-1}) \times \left(\sum_{S_{i-1} \rightarrow xS_i \in R} P(S_{i-1} \rightarrow xS_i) P_{rand}(x) \right) \times \left(1 + \sum_{S_i \rightarrow xS_i \in R} P(S_i \rightarrow xS_i) P_{rand}(x) \right),$$

where R is the set of production rules of G_{reg}, and $P_{rand}(x)$ is the probability of the occurrence of base x in the genome sequence. $OB(S_i)$ is the value of OB corresponding to the set of paths from an initial state to the state S_i. Thus, $OB = OB(S_{end})$ holds for end state S_{end} of G_{reg}. The time complexity for building OB is $O(n)$. In order to solve this minimization problem, we use a nonlinear optimization solver CFSQP [14]. Note that the probabilities $P(S_{i-1} \rightarrow xS_i)$'s are variables in this optimization problem. In conclusion, we can improve stochastic parameters by solving the following minimization problem:

Minimize:

$$OB = \sum_{w \in L(G_{reg})} Pr(w \mid G_{reg}) Pr(w \mid G_{rand}).$$

Subject to:

set of constraints in CN.

4. Experimental Results

The purpose of this paper is to show the effectiveness of the proposed approximation method for annotating RNA family in genome sequences. In this section, we will show some experimental results on how such approximation grammars can be used to reduce drastically the candidate positions of the RNA families.

The procedure of each experiment is given as follows:

(1) Get alignment data from the Rfam database [26].

(2) Generate a TAG_{RNA} G_1 that models RNA family by the system developed by Takakura [28].

(3) Generate an SREG G_2 approximating G_1 by the proposed method, whose time

Table 1 Experimental Results: The column Name is an RNA family name. The column ID is accession number in Rfam database. The value Length is the length of genome sequence. The column Start shows the position where the RNA family appears. The column cp gives the frequency of candidate positions. The value ratio is defined as cp/Length, i.e., filtering efficiency. Each column step i ($i = 2, 3, 4, 5$) shows the execution time of each step. Est. TAG_{RNA} column shows rough estimations of the execution time by a TAG_{RNA}. In each ID, the upper row is the result of optimized SREG, the lower is the result without optimization.

Name	ID	Length	Start	End	cp	ratio	Step 2	Step 3	Step 4	Step 5	est. TAG_{RNA}
Corona_FSE	AF356822	7730	6806	6892	4	0.05 %	0.11s	0.16s	4.7s	200m43s	1274days
					5352	69.2 %			1.1s	198m10s	
	M22457	7825	6904	6990	21	0.27 %		0.14s	4.5s	201m5s	1290days
					5409	69.1 %			1.1s	199m33s	
	X74312	7861	6937	7023	4	0.05 %		0.16s	5.1s	200m3s	1296days
					5318	67.7 %			1.1s	202m10s	
Corona_pk3	AY548235	3068	2972	3034	26	0.85 %	0.11s	0.09s	3.9s	31m28s	81days
					2773	90.4 %			1.1s	31m32s	
	M14878	1998	1781	1843	26	1.3 %		0.12s	3.8s	20m15s	52days
					1904	95.3 %			1.1s	20m13s	
	X0090	1767	1511	1572	15	0.85 %		0.09s	3.5s	17m55s	45days
					1708	96.7 %			1.1s	17m59s	
Entero_oriR	AF268065	7406	7281	7400	53	0.72 %	0.14s	0.27s	1m47.1s	502m40s	12145days
					7159	96.7 %			1.2s	503m13s	
	M33854	7399	7279	7399	63	0.85 %		0.23s	1m55.3s	501m13s	12131days
					7177	97.0 %			1.2s	501m53s	
	X79047	7501	7303	7423	66	0.88 %		0.23s	2m9.9s	509m05s	12301days
					7352	98.0 %			1.3s	512m40s	
HDV_rybozyme	AF309420	1676	685	773	38	2.3 %	0.14s	0.19s	18.8s	44m19s	306days
					1597	95.3 %			1.3s	44m28s	
	D01075	1682	686	774	41	2.4 %		0.16s	16.5s	44m35s	307days
					1602	95.2 %			1.3s	44m29s	
	M21012	1679	684	772	41	2.4 %		0.17s	16.1s	43m54s	307days
					1609	95.8 %			1.2s	44m37s	
Tombus_3_IV	AJ607402	4744	4652	4744	44	0.92 %	0.14s	0.14s	10.9s	128m23s	1066days
					4633	97.7 %			1.2s	124m32s	
	M25270	4701	4613	4701	43	0.91 %		0.18s	9.8s	125m58s	999days
					4551	96.8 %			1.2s	126m0s	
	X51456	2200	2110	2200	43	2.0 %		0.17s	11.3s	58m6s	466days
					2122	96.5 %			1.2s	58m35s	
Tymo_tRNA-like	AF035633	1379	1297	1379	32	2.3 %	0.12s	0.19s	13.1s	22m4s	335days
					879	63.7 %			1.2s	36m5s	
	D30753	2962	2879	2962	31	1.0 %		0.14s	10.6s	79m19s	999days
					2275	76.8 %			1.2s	79m27s	
	S97776	1255	1173	1254	35	2.8 %		0.17s	12.5s	32m31s	298days
					1087	86.6 %			1.2s	32m36s	

complexity for building an SREG is $O(n)$, where n is the size of G_1.

(4) Tune probabilistic parameters of G_2 by using nonlinear programming method, for which we use a solver CFSQP [14].

(5) Filter RNA sequences by using the SREG G_2, whose time complexity is $O(lm^2)$, where l is the length of total genome sequence, and m is the size of G_2.

(6) Check how much portion of candidate positions are remained.

The purpose of these experiments is to show efficiency and accuracy of our method, not to find new members of target RNA families. We used six RNA families including pseudoknots for this experiment:Corona_FSE, Corona_pk3, Entero_oriR, HDV_rybozyme, Tombus_3_IV, Tymo_tRNA-like. **Table 1** shows RNA data and execution time statistics of these experiments.

In order to emphasize on the importance of tuning stochastic parameters, and show the effectiveness of the method proposed in Subsection 3.4, for a given stochastic TAG, we generated two approximation grammars, one of which has stochastic parameters after optimization, and the other has parameters without optimization☆. The experimental results are sum-

☆ Non optimized parameters are obtained by finding feasible solutions (satisfying CN) using the software CFSQP.

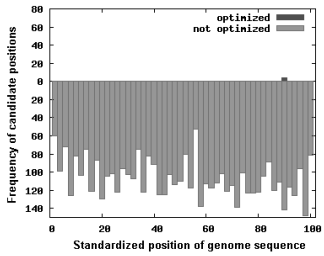

Fig. 7 Corona_FSE.

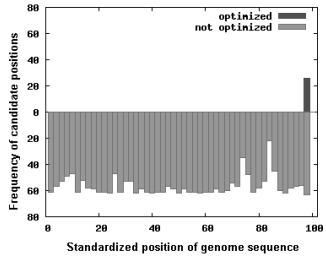

Fig. 8 Corona_pk3.

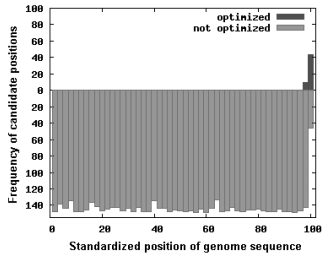

Fig. 9 Entero_OriR.

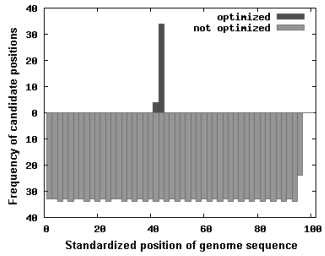

Fig. 10 HDV_ribozyme.

marized in **Figs. 7**, **8**, **9**, **10**, **11**, and **12**. The horizontal axis represents standardized position of RNA sequence in a RNA family, where 1 and 100 correspond to the start and the end of the sequence. More precisely, for an RNA sequence of length n, the horizontal value i shows the sequence segment approximately from the base $((i-1)\cdot n)/100$ to the base $(i\cdot n)/100$. The corresponding vertical value shows the number of positions in that segment found to be the candidates after the filtration.

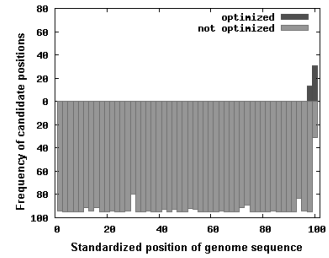

Fig. 11 Tombus_3_IV.

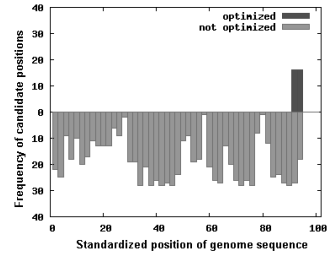

Fig. 12 Tymo_tRNA-like.

The experiments show that SREGs with optimized parameters performs much better than those with non-optimized parameters, which shows the effectiveness of the proposed method in Subsection 3.4. The filtering ratios with optimized SREGs show that the proposed method can have potential ability to drastically reduce the total time for annotating ncRNA families.

5. Related Works

There are so many algorithms proposed for predicting RNA secondary structures including pseudoknots for a given linear RNA sequence by Akutsu [1], Rivas [22], Reeder [20], Ren [21], Ruan [23], Uemura [31], etc. Although these standard prediction algorithms provide the basis for the analysis of given RNA sequences, they can not be directly used to find the location of a target RNA family in a complete genome sequence, which is the coverage of this paper.

Grammatical models are quite useful tools for improving the accuracy of predicting an RNA family location [7),11)~13),22),24),28),31)]. This paper proposed a grammatical technique for speeding up such analysis by incorporating the approximation theory of formal grammars. In this sense, in order to improve the accuracy of the analysis, it is important to obtain statistical information of a target RNA family. Thus, we need a reliable alignment data of RNA secondary structures. Recent advances in alignment algorithms of RNA secondary structures including pseudoknots will provide such a reli-

able alignment data [6),9),15),27)].

The most related is by Weinberg and Ruzzo, who proposed a method of approximating stochastic context free grammar by stochastic regular grammar and applied it to faster genome annotation of non-coding RNA families [32),33)]. Although CM method is quite effective for modeling and predicting RNA families, it can not model pseudoknotted structures. This work extends Weinberg and Ruzzo's approach to the case of stochastic TAGs by which we can model pseudoknots.

There are several grammar models proposed other than TAGs for modeling RNA secondary structures including pseudoknots, RNA Pseudoknot Grammars [22)], Multiple Context Free Grammars [11),12)], CFG based Parallel Communicating Grammars [3)], etc. As far as the authors' knowledge, the current work is the first attempt of applying approximation method to the grammars capable of modeling pseudoknots. The experimental results of this work suggest that similar approximation method might be quite effective also for the grammars other than TAGs.

Weinberg and Ruzzo's method has two steps. In the first step, they apply regular approximation method to stochastic TAGs. Second, they improve stochastic parameters. The approximation method in the first step proposed by them is almost equivalent to Nederhof's method [17)]. Nederhof proposed approximation of CFG by REG based on Recursive Transition Network. Furthermore, we know that there are some works on approximating TAGs by CFGs [8),19)]. Based on these works and Nederhof's works, we can approximate a given TAG by a CFG, and then approximate the obtained CFG by a REG. But, in the current paper, we apply regular approximation technique similar to Nederhof's method directly to TAGs for its simplicity.

In the second step, we improve stochastic parameters. The tuning method in the second step is different from Nederhof's method [18)]. Nederhof proposed a method for training finite automaton on SCFG so that Kullback-Leibler distance between them could be minimal. But, we do not use Kullback-Leibler distance for tuning probability parameters, since its direct application might loose some candidate locations of an RNA family which can be found by TAG_{RNA}. We use two grammar models G_{reg} and G_{rand} and apply nonlinear programming method in order to optimize parameters of G_{reg} with the guarantee that we do not loose such locations found by TAG_{RNA}. It might be a theoretically interesting future research topic to extend Kullback-Leibler distance minimization method so that it might not loose candidate locations which can be found by the target grammar.

6. Conclusions and Future Works

Inspired from the work by Weinberg and Ruzzo [32),33)], we developed a method for approximating a given stochastic tree adjoining grammar to stochastic regular grammar and applied it to faster genome annotation of ncRNA families. Parameter tuning based on an optimization technique is applied in order to improve the filtering efficiency. Preliminary experimental results were reported and the effectiveness of the proposed method was verified by these experiments.

Although we succeeded in filtering out non candidate positions of RNA families effectively, the parsing time efficiency of TAG itself is still computationally heavy for the practical use. In actuality, such parsing time inefficiency is currently a big research obstacle against the attempt of applying these mildly context sensitive grammars to RNA structure modeling. But, after the experience of these experiments, the authors believe that this kind of grammar approximation method would be quite effective also for the efficient parsing itself. We are now developing such an approximation method for stochastic TAG parsing.

Acknowledgments We would like to thank an anonymous reviewer of LATA'2007 for pointing out the relationship between this work and Nederhof's work.

References

1) Akutsu, T.: Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots, *Discrete Applied Mathematics*, Vol.104, pp.45–62 (2000).
2) Asakawa, H.: Parsing Tree Adjoining Grammar with Structural Information, Master's thesis, Dept. of Comp. Sci., Univ. of Electr.-Communi. (2004). (in Japanese).
3) Cai, L., Malmberg, R.L. and Wu, Y.: Stochastic Modeling of RNA Pseudoknotted Structures: A Grammatical Approach, *Bioinformatics*, Vol.19, suppl.1, pp.66–73 (2003).
4) Cannone, J.J., Subramanian, S., Schanare,

M.N., Collett, J.R., D'Souza L.M., Du, Y., Feng, B., Lin, N., Madabusi, L.V., Muller K.M., Pande, N., Shang, Z., Yu, N. and Gutell, R.R.: The comparative RNA Web (CRW) Site: An online database of comparative sequence and structure information for ribosomal, intron, and other RNAs, *BioMed Central Bioinformatics*, Vol.3, No.2 (2002).
5) Chen, J.L. and Greider, C.W.: Functional Analysis of the Pseudoknot Structure in Human Telomerase RNA, PNAS, Vol.105, pp.8080–8085 (2005).
6) Dost, B., et al.: Structural alignmnent of pseudoknotted RNA, LNCS 3903, pp.143–158 (2006).
7) Eddy, S.R. and Durbin, R.: RNA sequence analysis using covariance models, Nucleic Acids Res., Vol.22, No.11, pp.2079–2088 (1994).
8) Harbusch, K.: An Efficient Parsing Algorithm for Tree Adjoining Grammars, *Proc. 28th ACL*, pp.284–291 (1990).
9) Jiang, T., et al.: A general edit distance between RNA structures, *Proc. 5th a. Int. Conf. Comput. Molec. Biol.* (*RECOMB'01*), pp.211–220 (2001).
10) Joshi, A.K., Levy, L.S. and Takahashi, M.: Tree adjunct grammars, *J. Comput. Syst. Sci.*, Vol.10, pp.136–163 (1975).
11) Kato, Y., Seki, H. and Kasami, T.: On the Generative Power of Grammars for RNA Secondary Structure, *IEICE Trans. Inf. & Syst.*, Vol.E88-D, No.1, pp.53–64 (2005).
12) Kato, Y., Seki, H. and Kasami, T.: RNA Pseudoknotted Structure Prediction Using Stochastic Multiple Context-Free Grammar, *IPSJ Trans. Bioinformatics*, Vol.47, No.SIG17(TBl01), pp.12–21 (2006).
13) Kobayashi, S. and Yokomori, T.: Modeling RNA Secondary Structures Using Tree Grammars, *Proc.5th Genome Informatics Workshop*, pp.29–38 (1994).
14) Lawrence, C., Zhou, J.L. and Tits, A.L.: User's guide for CFSQP version 2.5: A C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints, Technical report, Institute for Systems Research, University of Maryland, College Park, TR-94-16rl (1997).
15) Matsui, H., et al.: Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures, *Bioinformatics*, Vol.21, No.11, pp.2611–2617 (2005).
16) Namy, O., Moran, S.J., Stuart, D.I., Gilbert, R.J.C. and Brierley, I.: A Mechanical Explanation of RNA Pseudoknot Function in Programmed Ribosomal Frameshifting, *Nature*, Vol.441, No.11, pp.244–247 (2006).
17) Nederhof, M.-J.: Regular Approximations of CFLs: A Grammatical View, *Proc. International Workshop on Parsing Technologies*, pp.159–170 (1997).
18) Nederhof, M.-J.: A General Technique to Train Language Models on Language Models, *Computational Linguistics*, pp.173–185 (2005).
19) Poller, P. and Becker, T.: Two-step TAG Parsing Revisited, *Proc. TAG+4*, pp.143–146 (1998).
20) Reeder, J., et al.: Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics, *BMC Bioinformatics*, Vol.5, No.104 (2004).
21) Ren, J., et al.: HotKnots: heuristic prediction of RNA secondary structures including pseudoknots, *RNA*, Vol.11, No.10, pp.1494–1504 (2005).
22) Rivas, E. and Eddy, S.R.: The language of RNA: A formal grammar that includes pseudoknots, *Bioninformatics*, Vol.16, No.4, pp.334–340 (1999).
23) Ruan, J., et al.: An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots, *Bioinformatics*, Vol.20, No.1, pp.58–66 (2004).
24) Sakakibara, Y., Brown, M., Hughey, R., Mian, I.S., Sjolander, K., Underwood, R.C. and Haussler, D.: Stochastic context-free grammars for tRNA modeling, *Nucleic Acids Res.*, Vol.22, pp.5112–5120 (1994).
25) Sakakibara, Y.: Pair hidden Markov models on tree structures, *Bioinformatics*, Vol.19, suppl.1, pp.232–240 (2003).
26) Sam, G.J., Simon, M., Mihairi, M., Ajay, K., Sean, R.E. and Alex, B.: Rfam: Annotating noncoding RNAs in complete genomes, *Nucleic Acids Res.*, Vol.33, D121–D124 (2005).
27) Seki, S. and Kobayashi, S.: A grammartical approarch to the alignment of structure-annotated strings, *IEICE Tran. Inf. & Syst.*, Vol.E88-D No.12, pp.2727–2737 (2005).
28) Takakura, T.: Automatic acquisition of grammatical models from RNA secondary structure alignment, Master's thesis, Dept.of Comp. Sci., Univ. of Electr.-Communi. (2005). (in Japanese).
29) Takakura, T., Asakawa, H., Seki, S. and Kobayashi, S.: Efficient Tree Grammatical Modeling of RNA Secondary Structures Including Pseudoknots, *Proc. 9th Annual International Conference on Research in Computational Molecular Biology*, Poster Abstracts, pp.339–340 (2005).
30) Uemura, Y., Hasegawa, A., Kobayashi, S. and Yokomori, T.: Grammatically Modeling and

Predicting RNA Secondary Structures, *Proc. 6th Genome Informatics Workshop*, pp.67–76 (1995).
31) Uemura, Y., Hasegawa, A., Kobayashi, S. and Yokomori, T.: Tree adjoining grammars for RNA structure prediction, *Theoretical Computer Science*, Vol.210, pp.277–303 (1999).
32) Weinberg, Z. and Ruzzo, W.L.: Faster Genome Annotation of Non-coding RNA Families Without Loss of Accuracy, RECOMB 2004, *Proc. Eighth Annual International Conference on Recerch in Computational Molecular Biology*, pp.243–251 (2004).
33) Weinberg, Z. and Ruzzo, W.L.: Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy, *ISMB/ECCB* (*Supplement of Bioinformatics*), pp.334–341 (2004).

(Received June 22, 2007)
(Accepted August 7, 2007)

(Communicated by *Hideo Bannai*)

Kazuya Ogasawara received B.E. degree from University of Electro-Communications in 2006. He is currently a student pursuing a Master of Computer Science at University of Electro-Communications. His research interests include formal language theory, and bioinformatics.

Satoshi Kobayashi has been Associate Professor of Department of Computer Science, University of Electro-Communications since 2000. He received the B.E., M.E., and D.E. degrees from the University of Tokyo in 1988, 1990, and 1993, respectively. His research interests include computational learning theory, formal language theory, theory of molecular computing, and bioinformatics.