Grammatically Modeling and Predicting RNA Secondary Structures

Yasuo UEMURA
uemura-y@gold.cs.uec.ac.jp

Aki HASEGAWA
haseg-ak@gold.cs.uec.ac.jp

Satoshi KOBAYASHI
satoshi@cs.uec.ac.jp

Takashi YOKOMORI
yokomori@cs.uec.ac.jp

Department of Computer Science and Information Mathematics,
University of Electro-Communications
1-5-1, Chofugaoka, Chofu, Tokyo 182, Japan

Abstract

Tree Adjunct Grammar for RNA (TAG^2_{RNA}) is a new grammatical device to model RNA secondary structures including pseudoknots. An efficient parsing algorithm for this grammar is developed, and applied to some computational problems concerning RNA secondary structures. With this parser, we first try to predict secondary structures of RNA sequences which are known to form pseudoknots structures, and show prediction results which nicely match the known structures. Further, a (−1) frameshift grammar is constructed based on a biological observation that a (−1) frameshift might be caused from some structural features of RNA sequences. The proposed grammar is used to find candidate sequences for (−1) frameshift in Human spumaretrovirus gag and pol genes.

1 Introduction

In recent study on the functional and structural properties of an RNA molecule, it turned out that some of the typical substructures play a crucial role in spacial folding of RNA sequences. Those include "hairpin (or stem-loop)", "bulge loop", and "interior loop" structures. Among others, the "pseudoknots" structure is today considered as one of the most typical and important structures found in RNA sequences ([2]). In reflecting on these perspectives of recent development in molecular biology, RNA secondary structure prediction problem is recognized

上村泰央、長谷川亜樹、小林聡、横森貴：電気通信大学情報工学科，〒 182 東京都調布市調布ヶ丘 1-5-1

as one of the most fundamental but attractive problems to be solved. Because of its difficulty in nature, however, almost all the existing approaches to this steep cliff work can only handle the secondary structures not containing pseudoknots. This is a critical defect in predicting the secondary structure of an RNA molecule.

In our previous paper [6], a new grammatical device called tree adjunct grammars has been proposed to model the RNA secondary structures including "pseudoknots", and the appropriateness of the modeling grammars has been demonstrated by showing several modeling examples on biological data in such as HIV-2 gag-pol Region (Figure 1), Group I Intron Core Region, and 16S rRNA.

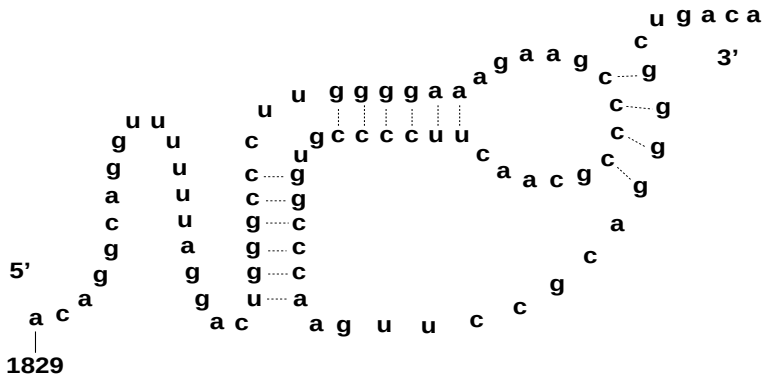

Figure 1: A Typical RNA Secondary Structure in HIV-2 gag-pol Region

In this paper, we present an efficient parsing algorithm for our tree grammar TAG^2_{RNA} which actually runs in time polynomial in the input size n (or more exactly in time $O(n^4)$). This high feasibility leads us to constructing a practical software tool which, for a given RNA sequence, produces its secondary (parsing) structure and predicts an unknown secondary structure. In fact, we have built up such a software system and made several experiments on RNA sequence data. We first try to predict secondary structures of RNA sequences which are known to form pseudoknots structures, and show prediction results which nicely match the known structures. Further, a (−1) frameshift grammar is constructed based on a biological observation that a (−1) frameshift might be caused from some structural features of RNA sequences. The proposed grammar is used to find candidate sequences for (−1) frameshift in Human spumaretrovirus gag and pol genes.

The rest of the paper is constructed as follows. Section 2 briefly introduces the modeling grammar TAG^2_{RNA} and describes its relation to the prediction problem of RNA secondary structures. Section 3 gives the method for predicting RNA secondary structures. Experimental results obtained from RNA sequence data are presented in Section 4. Section 5 concludes the paper with related works and a future perspective of this research.

2 Modeling Grammars

In this section, we informally define tree adjunct grammars for RNA (TAG^2_{RNA}) proposed in [6]. This is a valiant of Tree Adjunct Grammar which is originally introduced by Joshi et al. ([5]).

Then, we discuss how this grammar models RNA secondary structures, and how prediction problem of RNA secondary structures are formalized in our framework. We also discuss the language class of the grammar.

2.1 Definitions

Let $\Sigma = \{\mathtt{a}, \mathtt{u}, \mathtt{g}, \mathtt{c}\}$ be a finite alphabet, each of whose element represents a base in RNA sequences. For representing the Watson-Crick base pairing, we use a bar notation: $\overline{\mathtt{a}} = \mathtt{u}$, $\overline{\mathtt{c}} = \mathtt{g}$, $\overline{\mathtt{g}} = \mathtt{c}$, $\overline{\mathtt{u}} = \mathtt{a}$. TAG^2_{RNA} is a triple $G = (C, A, F)$, where C, A and F are the set of center trees, the set of adjunct trees, and the set of final symbols, respectively. Center and adjunct trees are called *elementary trees* of the grammar. All types of elementary trees allowed in TAG^2_{RNA} are shown in Figure 2, where X,Y,Z represent nonterminal symbols, x represents a terminal symbol ($x \in \Sigma$), and λ denotes an empty string. A center tree is restricted to only TYPE 1, that is, C is a set of TYPE 1 trees and each element of C has different nonterminal symbol at its root node. On the other hand, an adjunct tree is taken from TYPE 2 to TYPE 5. Note that every adjunct tree has both its root and exactly one leaf node, called a *foot* node, labeled by the same nonterminal symbol.

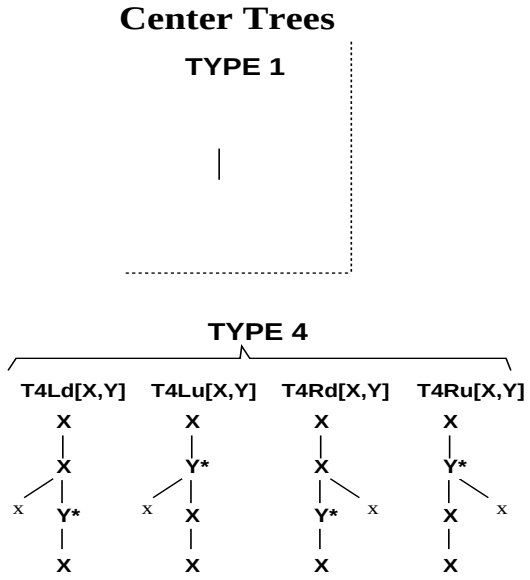

Figure 2: The Types of Elementary Trees Allowed in TAG^2_{RNA}

As is shown in Figure 2, each type of tree consists of internal nodes and leaf nodes, where internal nodes are labeled by nonterminal symbols, and leaf nodes are labeled by terminal or nonterminal symbols except TYPE1. A node labeled by a nonterminal symbol is called a *nonterminal node*. Note that some nonterminal nodes are tagged with an additional notation $*$, which indicates the position where we can apply an *adjunction* operation defined below.

An adjunction operation composes trees of the grammar as follows. Let γ be a tree containing a tagged node labeled by nonterminal symbol X and let β be an adjunct tree with the root

labeled by the same symbol X. Then, we say β is *adjoinable* to γ, and the resulting tree by adjoining β to γ is shown in Figure 3. That is, the subtree of γ whose root is the tagged node labeled by X is first removed from γ, then β is inserted in its place, and finally the subtree is attached to the foot node of the adjunct tree β. We say γ_1 is a *derived tree* of γ_2, if and only if either $\gamma_1 = \gamma_2$ or γ_1 can be obtained by successively adjoining a sequence of adjunct trees in A to γ_2.

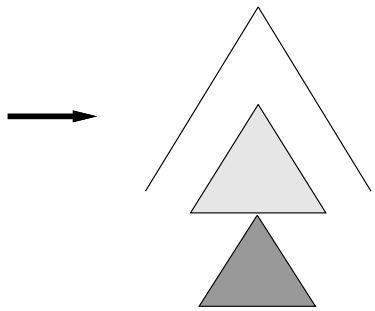

Figure 3: An Adjunction Operation

We now define the tree set of G as the set of all derived trees of center trees in C, and denote it by $T(G)$. A tree in $T(G)$ whose tagged nodes are all labeled by final symbols is called an *acceptable* tree of G. By $AT(G)$, we denote the set of all acceptable trees of G. Then, the language generated by G is defined as the set of all terminal strings which appear in the *frontier* of the trees in $AT(G)$, where frontier means left-to-right ordered sequence of leaf nodes of the tree. Note that any nonterminal symbol does not appear in the frontier of $T(G)$.

2.2 Modeling and Predicting RNA Secondary Structures

Now we describe how TAG^2_{RNA} models RNA secondary structures. For any RNA sequence w, we can regard the *secondary structure* of w as a set of tuples (p,q) with $p < q$ which represents a base pairing between p-th and q-th bases of w. On the other hand, for given G and w, let γ be a tree in $AT(G)$ whose frontier has the labels equal to w. Then, the secondary structure of w is defined as the set of all (p,q) such that p-th and q-th symbols of w are generated by exactly one adjunct tree of TYPE 2 or TYPE 3 during the derivation of γ, where p-th and q-th just correspond to x and $\overline{x}$ in the adjunct tree, respectively. Thus, a tree in $AT(G)$ can model a secondary structure. The following example illustrates the relationship between a derived tree and a secondary structure.

Example 1 In Figure 4, T is a derived tree which generates the string "`agacuu`" and represents the secondary structure $\{(1,5),(2,4),(3,6)\}$. Note that the structure represented by T has crossing dependencies which cannot be modeled by any CFG.

Generally, several trees in $AT(G)$ may represent the same string w. This means that there are several structures for an RNA sequence w to be able to form. Therefore, it is important issue

Figure 4: A Derived Modeling the Secondary Structure $\{(1,5),(2,4),(3,6)\}$

to choose one from such structures. In many other works on the prediction of RNA secondary structures, the problem is defined to be a search for an optimal structure with respect to some evaluation function based on free energy theory. Following this, we define an evaluation function f which is used to choose an optimal structure in our work. (Section 3.2)

Now we formalize the prediction problem of RNA secondary structures as follows: given an RNA sequence w, find a possible set S of address pairs of w such that $f(S)$ marks the best score. In fact this can be achieved by parsing w using our tree grammars. Thus, one may claim that prediction is nothing but parsing in our framework.

2.3 TAG^2_{RNA}'s Generative Capability

Note that a prediction result strictly depends on G which is given to the parser, since the parser can output only the structures represented by G. So it is an important issue to construct G in the prediction process. From observing and examining some biological example data, we chose TAG^2_{RNA} for the underlying grammar G in this paper. (It would be an important open problem to learn G from example data.)

What kinds of secondary structures can be represented by TAG^2_{RNA} is theoretically unclear in the current status of this work. But we insist that TAG^2_{RNA}'s are more powerful than CFG's, which can be easily proved by the existence of an effective algorithm transforming any CFG into a TAG^2_{RNA}. Furthermore, this extra power is important to represent RNA secondary structures, because it can elegantly capture reversal crossing dependecies which appear in pseudoknots structures. In fact, we observe that any example in the actual biological sequences can be modeled by our grammar.

3 Method

We first design an efficient parsing algorithm for our grammar TAG^2_{RNA} and implement the parsing algorithm. Then, taking an evaluation function f into consideration, we develop a

software tool for predicting RNA secondary structures.

3.1 Parsing Algorithm

Our parsing algorithm is based on the algorithm for Tree Adjoining Grammars by Vijay-Shankar and his coworker ([8]). The time complexity of their algorithm is $O(n^6)$. Although our parsing algorithm has some similarities to theirs in that TAG^2_{RNA} can deal with trees generating empty strings (TYPE 5), which are not allowed in Vijay-Shankar's TAG. Now we describe the features of the parsing algorithm.

- TAG^2_{RNA} parser is a bottom-up parsing algorithm in nature.
- TAG^2_{RNA} parser uses four-dimensional dynamic programming method.
- TAG^2_{RNA} parser can find an optimum solution with respect to some evaluation functions.
- The time and space complexity is $O(n^4)$, where n is the length of an input string.

The time complexity $O(n^4)$ is achieved under the constraint that TYPE 5 adjunct trees must be applied before every other type of adjoining trees. But in general case, the time complexity is $O(n^5)$.

Given a string $w = a_1 a_2 \cdots a_n$ and a tree grammar $G = (C, A, F)$, the parsing algorithm computes four-dimensional matrix $B[i,j,k,l]$ $(0 \leq i \leq j \leq k \leq l \leq n)$ of sets of elements in A such that $\gamma \in B[i,j,k,l]$ if and only if their exists a derived tree of γ whose frontier is $a_{i+1} \cdots a_j X a_{k+1} \cdots a_l$, and every tagged node is labeled by a final symbol. Then, it searches $B[0,j,j,n]$ $(0 \leq j \leq n)$ for γ which is adjoinable to a center tree $\alpha \in C$. It holds that $w \in L(G)$ if and only if such a γ exists.

3.2 Prediction Method

Our method for predicting RNA secondary structures consists of the following procedures:

1. Set up the most general grammar $G_0 = (C, A, \{S\})$, where every nonterminal nodes are labeled by exactly one symbol S, and A contains all types of adjunct trees.
2. Construct an evaluation function f based on the free energy table (Table 1) proposed in [9].
3. Parse a given string w with G_0 where the evaluation function f is used in order to compute the total free energy of each parsed tree.
4. Choose one parsed tree which has the minimum value of free energy.

The above method provides one of the most stable structures for w modeled by TAG^2_{RNA}, and as we have previously mentioned, the obtained secondary structure is regarded as a prediction result. In this paper, we do not take the base pairing between `g` and `u` into consideration.

Table 1: Free Energy Table

Increased free energy per one base pair formation [kcal/mol]

Base Pairing on 3′ End	Base Pairing on 5′ End				
	GU	AU	UA	CG	GC
GU	-0.5	-0.5	-0.7	-1.5	-1.3
AU	-0.5	-0.9	-1.1	-1.8	-2.3
UA	-0.7	-0.9	-0.9	-1.7	-2.1
CG	-1.9	-2.1	-2.3	-2.9	-3.4
GC	-1.5	-1.7	-1.8	-2.0	-2.9

From Table 1, we can allocate free energy value to each stacked region. We compute free energy only from base pair formation, but not from loop formation, since it is difficult to compute the loop energy in the parsing process of our algorithm. Further, it should be noted that there is no established method for computing free energy of pseudoknots structures.

4 Experimental Results

In this section, we present the results of experiments which we have made with our prediction system.

4.1 Predicting RNA Secondary Structures

We first try to predict the secondary structures of RNA sequences which are known to form pseudoknots structures. In this experiment, we predict secondary structures of two RNA sequences of length 70 obtained from HIV-2 gag-pol region and 16S rRNA. The results are shown in Figure 5 where (1) a solid line indicates a known stacked region, (2) a dotted line links two subsequences which are predicted to form a stacked region by our method, more exactly, bold dotted and fine dotted denote stacked regions with high and low free energy, respectively.

Ignoring fine dotted lines, the prediction results nicely meet the known structures.

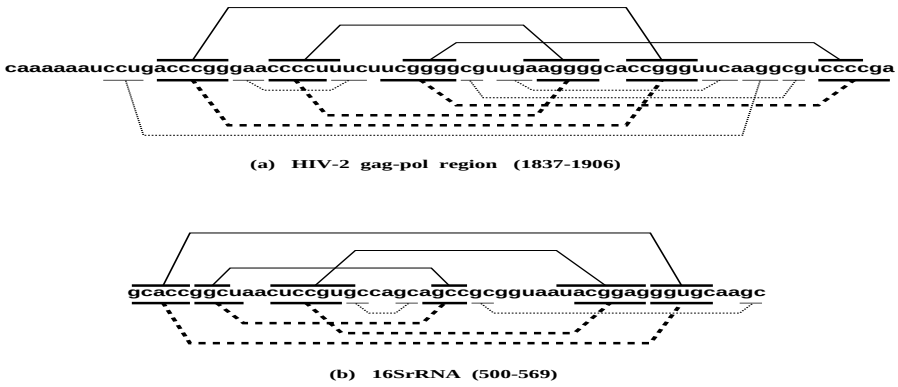

Figure 5: Observed Secondary Structures VS. Predicted Secondary Structures

4.2 Predicting Frameshift Regions

It is known that the existence of a shifty codon (e.g. UUUU and CCCC) or the formation of some characteristic secondary structures in mRNA might cause shifting of reading frame. In the case of retroviruses, a (-1) frameshift occurs at the shifty codon which is followed by a pseudoknots structure.

Based on biological observations of retroviruses, we determined characteristic features in (-1) frameshift regions as is shown in Figure 6. Then, we converted it into our grammar, which we call (-1) frameshift grammar. The feasibility of TAG^2_{RNA} allows us to construct a grammar which represents such a complicated structure.

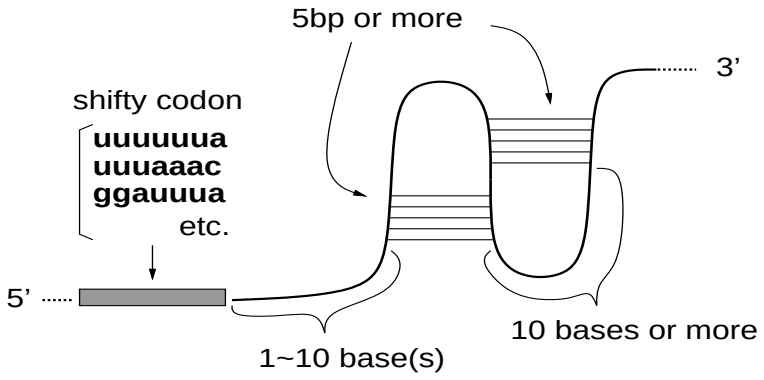

Figure 6: Characteristic Features Causing (-1) Frameshift

With this grammar, we search Human spumaretrovirus gag and pol genes for the subsequences which can be represented by the (-1) frameshift grammar, and have found three subsequences illustrated in Figure 7. These obtained structures are highly expected to cause (-1) frameshift.

5 Summary and Conclusions

Recent work on the stochastic prediction and identification of tRNAs secondary structures by Sakakibara et al. has close relationship to our works ([7]). They proposed stochastic version of context free grammars for modeling RNA secondary structures, and applied it to the prediction problem of tRNA secondary structures. However, since their stochastic grammar is based on CFG, it can not represent reversal crossing dependencies, which can be represented by our TAG^2_{RNA}. It is interesting to develop stochastic version of our grammar, which can be realized by distributing probabilities to adjunct trees.

Abe and Mamitsuka proposed an extended version of TAGs in stochastic configurations, called Stochastic Ranked Node Rewriting Grammars (SRNRG), for the purpose of predicting protein secondary structures including β sheet regions ([1]). As is shown in their works, SRNRG can represent more complicated non-planar secondary structures, which can not be described with TAG^2_{RNA}. However, in spite of this defect, TAG^2_{RNA} may have sufficient ability to represent

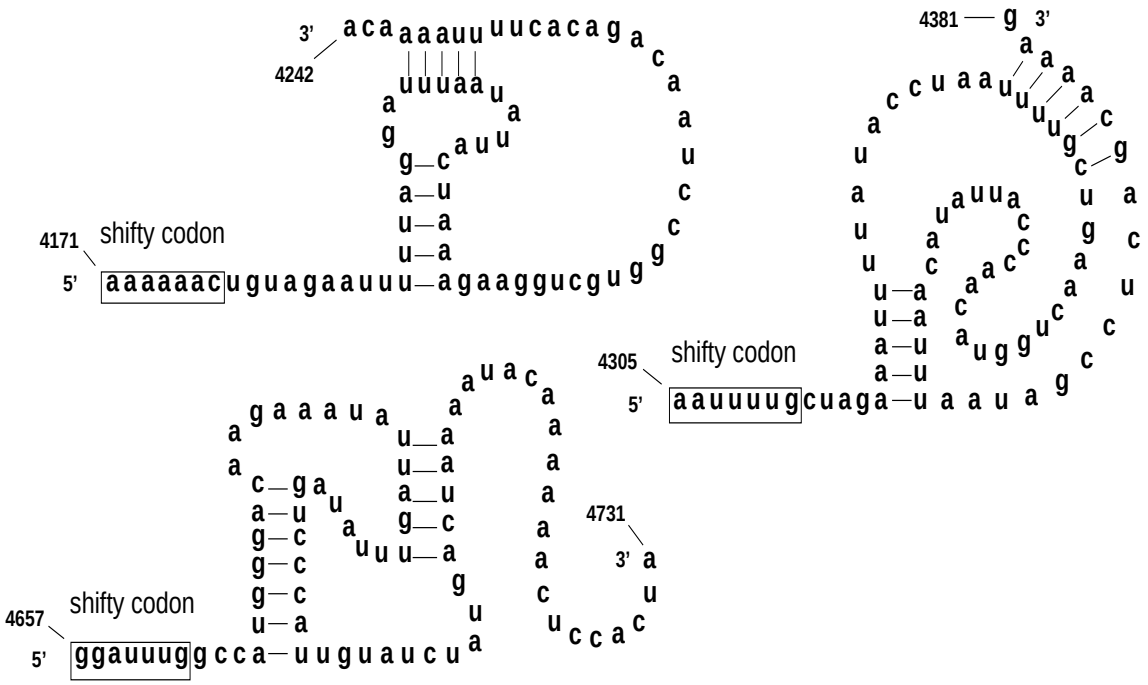

Figure 7: Candidates for (-1) Frameshift Region in Human spumaretrovirus gag and pol genes

RNA secondary structures. Further, it should be noted that the proposed parser is remarkably efficient compared to their parser.

In the current paper, we designed an efficient parsing algorithm for TAG^2_{RNA}, which is a modeling grammar for RNA secondary structures, and implemented the algorithm. Then, we made several experiments on RNA sequence data for modeling and predicting secondary structures. However, the proposed algorithm has a problem that the space complexity $O(n^4)$ makes it difficult to deal with long sequences. So we need to devise some methods for overcoming this difficulty.

The parsing algorithm is the core part of an integrated system assisting biologists in predicting and learning RNA secondary structures. In future works, we shall develop a method for learning TAG^2_{RNA} grammars from given RNA sequences, which could be used, for example, in order to obtain the (-1) frameshift grammar proposed in Section 4.2.

Furthermore, we need to provide some convenient user interface for putting RNA secondary structures into the system, since grammatically representing secondary structures would be a heavy task for those who are not familiar with the rudiments of the language theory. For this purpose, we must devise some graphical representations of secondary structures which can be automatically converted to our grammar.

Acknowledgements

This work was supported in part by Grants-in-Aid for Scientific Research No.07780310 and No.07249201 from the Ministry of Education, Science, Sports and Culture, Japan. We thank

P. Kaplan (NEC, New Jersey) and Y. Akiyama (Kyoto Univ.) for useful suggestions about prediction method based on free energy computation.

References

[1] N. Abe and H. Mamitsuka, "Prediction of Beta-Sheet Structures Using Stochastic Tree Grammars," In *Proceedings of Genome Informatics Workshop V*, Universal Academy Press, pages 19-28, 1994.

[2] E. Dam, K. Pleij and D. Draper, "Structural and Functional Aspects of RNA Pseudoknots," *Biochemistry*, Vol.31, no. 47, pages 11665-11676, 1992.

[3] T. Jacks, M. D. Power, F. R. Masiarz, P. A. Luciw, P. J. Barr, and H. E. Varmus, *Nature*, **331**, 280, 1988.

[4] T. Jacks, H. D. Madhani, F. R. Masiarz, and H. E. Varmus, *Cell*, **55**, 447, 1988.

[5] A. K. Joshi, L. S. Levy and M. Takahashi, "Tree Adjunct Grammars," *Journal of Computer and System Sciences*, vol.10, pages 136-163, 1975.

[6] S. Kobayashi and T. Yokomori, "Modeling RNA Secondary Structures Using Tree Grammars," In *Proceedings of Genome Informatics Workshop V*, Universal Academy Press, pages 29-38, 1994.

[7] Y. Sakakibara, M. Brown, R. C. Underwood, I. S. Mian and D. Haussler, "Stochastic Context-free Grammars for Modeling RNA," In *Proceedings of 27th Hawaii International Conference on System Sciences*, vol.V, pages 284-293, 1994

[8] K. Vijay-Shankar and A. K. Joshi, "Some Computational Properties of Tree Adjoining Grammars," In *Proceedings of 23rd Annual Meeting of the Association for Computational Linguistics*, pages 82-93, Chicago, IL.

[9] D. H. Turner, N. Sugimoto, J. A. Jaeger, C. E. Longfellow, S. M. Freier, and R. Kierzek, Cold Spring Harb. Symp. Quant. Biol., **52**, pages 123-133, 1987.