CORPUS-BASED LEARNING OF STOCHASTIC CONTEXT-FREE GRAMMARS COMBINED WITH HIDDEN MARKOV MODELS FOR tRNA MODELLING

Juan Miguel García-Gómez[1], **José Miguel Benedí**[2], **Javier Vicente**[1] **and Montserrat Robles**[1]
Informática Médica-BET[1] and Dpto. Sistemas Informáticos y Computación[2]
Univ. Politécnica de Valencia
Camino de Vera s/n, 46022, Valencia, SPAIN
{juanmig@upvnet, jbenedi@dsic, javirob@fis, mrobles@fis}.upv.es

Abstract:
In this paper, a new method for modelling tRNA secondary structures is presented. This method is based on the combination of Stochastic Context-Free Grammars (SCFG) and Hidden Markov Models (HMM). HMM are used to capture the local relations in the loops of the molecule (non-structured regions), and SCFG are used to capture the long-term relations between nucleotides of the arms (structured regions). Given annotated public databases, the HMM and SCFG models are learned by means of automatic inductive learning methods. Two SCFG learning methods have been explored. Both of them take advantatge of the structural information associated to the training sequences: one of them is based on a stochastic version of the Sakakibara algorithm and the other one is based on a Corpus-based algorithm. A final model is then obtained by merging of the HMM of the non-structured regions and the SCFG of the structured regions. Finally, the performed experiments on the tRNA sequence corpus and the non-tRNA sequence corpus give significant results. Comparative experiments with another published method are also presented.

Keywords: RNA; secondary structure modelling; language modelling; grammatical inference; stochastic context-free grammar (SCFG); Hidden Markov Models (HMM); syntactic pattern recognition.

Reference to this paper should be made as follows: García-Gómez, J.M., Benedí, J.M., Vicente, J., Robles, M. (2005) 'CORPUS-BASED LEARNING OF STOCHASTIC CONTEXT-FREE GRAMMARS COMBINED WITH HIDDEN MARKOV MODELS FOR tRNA MODELLING', *International Journal of Bioinformatics Research and Applications*, Vol. x, No. x, pp.xxx–xxx. 2

Biographical notes: Juan Miguel Garcia-Gomez received the Master degree in Computer Science from Politechnic University of Valencia. Since 2004, He has been with the Department of Applied Physics as an Associated Professor. His current research interest lie in the Pattern Recognition techniques applied to Bioinformatics and Decision Support Systems.

José-Miguel Benedí received the Licenciado degree in physics from the University of Valencia in 1980 and the Ph.D. degree in computer science

1 Introduction

The procurement of the secondary structure of biomolecules supports research in the understanding of biological processes and interactions. This work focuses on the modelling of the secondary structure of molecules with biological palindromes. The noncoding ribonucleic acid (ncRNA) families have this characteristic secondary structure. ncRNA is any RNA transcript that functions directly as RNA rather than being translated into protein. The list of ncRNAs is extensive and growing. They are a diverse collection, ranging in size from 21 nucleotides (miRNAs) to more than 10,000 nucleotides (Xist). In general, an expert who can model the grammar is required in order to obtain these models. Since, this is a very tedious task, is desirable to make unambiguous grammars with a small number of rules. In order to investigate and evaluate the possibility of automatically modelling the secondary structure of noncoding ribonucleic acids (ncRNA) molecules and help the understanding of undiscovered non-coding genes (as miRNA), the method proposed in this work attempts to model the secondary structure of the tRNA molecules.

The primary structure of the tRNA molecules is encoded by a linear string of four different constituent nucleotides: adenine (A), cytosine (C), guanine (G) and uracil (U) [3]. The string of the tRNA molecules contains the biological palindromes, which shape the secondary structure of tRNA [21, 23].

tRNA secondary structure is made up of arms and loops (Figure 1). There are four arms: the acceptor arm, the D arm , the T pseudouridine C arm, and the anticodon arm. These arms are the structured regions of the tRNA. There are four loops: the D loop, the T pseudouridine C loop, the anticodon loop, and the variable loop. These loops (together with the duplet composed by the 8th and 9th nucleotides) are the non-structured regions of the tRNA molecule.

This paper addresses tRNA secondary structure analysis under the perspective of stochastic modelling and syntactic pattern recognition. tRNA molecules can be considered as strings of discrete symbols with a hidden syntactic structure to be modelled (Figure 2) [15, 23].

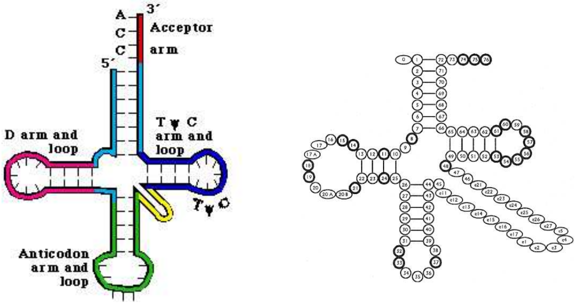

Figure 1: tRNA regions and tRNA schema [15, 23].

```
G G G G U A U   U A  G C U C     A G U U G G U A    G A G C
<AceptorArm>  < R0 > <D-Arm>     <   R1          >   <D-Arm>

G C A A C A     A U G G C A U    U G U U G A  G G U C
<AntiC-Arm>     <   R2      >    <AntiC-Arm>  <  R3 >

A G C G G    U U C G A C C    C C G C U  A U G C U C C
<   TYC >    <   R4      >    <  TYC  >  < AceptorArm>
```

Figure 2: tRNA primary and secondary structure.

tRNA modelling has been studied previously within the bioinformatics and the pattern recognition disciplines. In [12], Lowe and Eddy presented tRNAscan-SE, which is a prediction tool composed of several search steps. This system uses the tRNAscan program [6], an implementation of a multistep weight matrix algorithm for identification of eukaryotic tRNA promoter regions [17], and the RNA covariance analysis package Cove [5].

In [20], Sakakibara et al. introduced a model for estimating SCFG of the secondary structure for aligning tRNA sequences. The capability of the Sakakibara method for aligning and discriminating among tRNA groups was studied in [19].

In this paper, we propose a combination of SCFG and HMM to model the secondary structure of the tRNA molecules. The SCFG are used to represent the long-term relations of the structured regions, while the HMM are used to capture the local relations of the non-structured regions.

HMMs capture the local relations in the loops of the molecule (non-structured regions); SCFGs capture the long-term relations between nucleotides of the arms (structured regions). Given annotated public databases, the HMM and SCFG models are learned by means of automatic inductive learning methods. Two SCFG learning methods are explored. Both of them take advantatges of the structural information associated to the training sequences: one of them is based on a stochastic version of the Sakakibara algorithm, and the other is based on a Corpus-based algorithm. Models obtained from both proposals are reestimated by means of an estimation Earley-based algorithm. The HMM of the non-structured regions and the SCFG of the structured regions are finally merged in order to obtain the unified model of the tRNA secondary structure.

In order to evaluate the behaviour of this proposal, we carried out experiments with a tRNA sequence corpus and a non-tRNA sequence corpus. A comparison with the tRNAScan-SE system completes the evaluation of our method.

2 Methodology

The structured and non-structured regions of the tRNA secondary structure are modelled by SCFG and HMM, respectively. In this section we present the learning process of these models and the fusion method to obtain the final combined model.

First, we introduce the HMM notation and the SCFG notation that are used in this work. The Hidden Markov Model [14] (HMM) (Q, Σ, π, A, B) is a 5-tuple, where Q is a set of states; Σ is the finite set of terminal symbols (in RNA problem $\{A, U, C, G\}$); A is a matrix with the transition probability from one state to another, and B is a matrix with the emission probability of each symbol in each state.

A Context-Free Grammar [1] (CFG) (N, Σ, P, S) is a 4-tuple, where N is a finite set of non-terminal symbols; Σ is the finite set of terminal symbols where $N \cap \Sigma = \emptyset$; P is a finite set of rules of the form $A \rightarrow \alpha$, where $A \in N$ and $\alpha \in (N \cup \Sigma)^+$; and S is the initial symbol ($S \in N$). SCFG is the stochastic extension of CFG; G_s is a pair (G, p), where G is a CFG; and $p : P \rightarrow]0, 1]$ is a probability distribution over the grammar rules such that $\forall A \in N : \sum_{\alpha \in (N \cup \Sigma)^+} p(A \rightarrow \alpha) = 1$.

```
[ [ G [ G [ G [ G [ T [ A [ T [ R0 [ G [ C [ T [ C [ R1 ] G ]
A ] G ] C G [ C [ A [ A [ C [ A [ R2 ] T ] G ] T ] T ] G A
[ R3 [ A [ G [ C [ G [ G [ R4 ] C ] C ] G ] C ] T ] ] ] ] ]
A ] T ] G ] C ] T ] C ] C ] ]
```

Figure 3: tRNA sample bracketed and categorized.

Given an annotated structural corpus, the learning of the HMM and SCFG are carried out by means of automatic inductive learning methods presented in Section 2.1. Finally, the fusion procedure to produce the final combined model is also presented in Section 2.2.

2.1 Learning of the models

The specific models of the tRNA secondary structure are inductively learned from the annotated corpus, "Compilation of tRNA sequences" database [24]. This kind of corpus includes the alignment of the samples with the secondary structure schema. Therefore, there are multiple-alignments among the samples, and each nucleotide of a sample has a fixed position in the secondary structure (Figure 1).

The alignment with the secondary structure (Figure 2) allows us to easily extract the non-structured regions from the sequences and to replace them with category symbols. Therefore, we have the categorized input sentences available to learn the SCFG, and we have the samples of each one of the non-structured regions to learn their corresponding HMM.

HMM have proven to be useful in biological modelling problems in many tasks [8, 9, 13]. HMMs are simple and robust models, and both the estimation process of their parameters and the interpretation mechanism are well-known in the literature [14]. HTK, an HMM open source toolkit, was used to carry out the experiments presented in this paper [28]. Five models were learned for each one of the non-structured regions: the duplet composed of the 8th and 9th nucleotides (R0 region links the TYC arm and the D-arm); the D-loop (R1); the Anticodon loop (R2); the Variable loop (R3); and the TYC loop (R4). Different topologies were explored and a wide range of numbers of states were tested. The Baum-Welch algorithm was used to estimate the HMM parameters. Details of these experiments are presented in Section 3.

Thanks to the multiple-alignment of the samples with the secondary structure schema, we can associate each sequence with its structural tree so that the learning of the SCFG takes advantage of this information. In order to obtain initial models of the structured regions, two SCFG learning methods were explored: one based on a stochastic version of the Sakakibara algorithm [18, 19] and one based on Corpus-based algorithm [2]. Both of them make use of the structural information of the training corpus.

The Sakakibara algorithm infers a reversible CFG that is consistent with the corpus of bracketed samples (Figure 3). A CFG is said to be reversible if both the following hold:

1. The grammar is invertible, $A \rightarrow \alpha$ and $B \rightarrow \alpha$ in P implies $A = B$

2. The grammar is reset-free, $A \rightarrow \alpha B \beta$ and $A \rightarrow \alpha C \beta$ in P implies $B = C$

The first step of the Sakakibara algorithm creates context-free rules for every internal node of the trees in the samples. The training corpus alphabet is the terminal alphabet of the grammar. A merging process is then carried out to join the non-terminals that do not accomplish the invertible and the reset-free conditions. The invertible condition is eliminated by joining non-terminals whose rules have the same right-hand sides. The reset-free condition is eliminated by joining the non-terminals that appear in the right-hand side of rules. These rules must have the same left hand side and they must have the same symbols on the right hand side. The steps for eliminating the two conditions are repeated until no new merge is produced. Then the CFG of the reversible language that includes the sample is obtained [18]. Finally, a stochastic version of this algorithm allows us to obtain the initial probabilities of the SCFG models.

The second method to obtain the initial SCFG was inspired by previous works in Natural Language Models [2, 11]. Following these works, we infer the initial SCFG of the structured regions of the tRNA molecules taking advantage of the multiple alignments of the samples with the secondary structure schema. Therefore, the trees with labelled internal nodes offer the necessary information to the Corpus-based algorithm to obtain the SCFG by counting the rules of the full labelled trees. The Corpus-based algorithm input is a set of trees with labelled internal nodes and its results is the SCFG. The rules of the SCFG are those that appear in the set of trees. The probability of each rule is estimated by counting the appearance of each one in the trees. Then probability is normalized for each non-terminal.

Once the SCFGs of the structured regions are learned, a re-estimation process can be executed. The estimation algorithm: the Inner-Outer (IO), is based on the stochastic version of the Earley algorithm for SCFGs in general form [25, 11]. This algorithm can be optimized to be able to use bracketed samples to take advantage of the structural information of the samples (IOb) [11].

2.2 Fusion and analysis of the models

In order to analyze original tRNA sequences, a combined model from the models of structured and non-structured regions must be obtained.

HMM are converted to equivalent stochastic grammars in order to merge models of non-structured regions to the SCFG of the structured region. Then, the R0-R4 terminal symbols of the SCFG are sustituted by the initial symbols of equivalent stochastic grammars of the non-structured regions. Finally, a complete SCFG is obtained by fusion of the models.

The analysis of an original tRNA sequence is carried out by means the Earley algorithm [4]. The Earley algorithm processes sequences from left to right, filling lists of positions with items that indicate the application of the grammatical rules, the interpreted and non-interpreted parts of the consequent, and the link to the position in which the items have been inserted. A stochastic version of the

Table 1: Original training and testing corpora.

	Corpus	$card(X)$	$\mid \bar{X} \mid$	$s(\mid X \mid)$	Min	Max
T	tRNA+	3587	76.17	5.17	62	95
t	tRNA+	1323	76.12	5.16	67	93
t	LSU-	1323	72.53	10.52	36	96

Earley algorithm is the Inner algorithm, which computes the probability of the sample over the complete model [26, 10].

3 Experiments

This section presents our experimental work using the proposed algorithms with real tRNA samples. The experiments that were carried out to evaluate the method use the "Compilation of tRNA sequences and sequences of tRNA genes" [24].

Table 1 shows an abstract of training and testing corpora. In this table: $card(X)$ is the cardinality; $\mid \bar{X} \mid$ is the average length; $s(\mid X \mid)$ is the length standard deviation; Min is the minimum length and Max is the maximum length. T represents the training corpus and t represents the test corpus. The tRNA+ represents the positive samples (real tRNA sequences) and the LSU- represents the negative samples (non-tRNA sequences).

The negative test corpus was prepared from the LSU-RRNA database [27]. Given that the tRNA sample lengths varied from 67 to 93 nucleotides, similar lengths in negative test samples were required. A random procedure to extract disjoint sub-sequences from LSU-RRNA samples was carried out.

In order to learn the structured regions and the non-structured regions separatively, the categorization process described above was applied. Table 2 shows an abstract of the training corpora for structured and non-structured regions. The tRNA*+ is the categorized positive corpus of the structured regions, and R0-R4 are the corpora of non-structured regions. The statistics of the categorized positive test are shown in table 2 to evaluate the precision and recall rates of the model of the structured regions.

Table 2: Processed training and testing corpora.

	Corpus	$card(X)$	$\mid \bar{X} \mid$	$s(\mid X \mid)$	Min	Max
T	tRNA*+	3587	50.73	1.69	37	54
t	tRNA*+	1323	50.73	1.69	37	54
T	R0	3587	1.99	0.05	1	2
T	R1, D	3587	8.45	0.94	6	11
T	R2, Anti	3587	7	0	7	7
T	R3, Var	3587	6.20	4.60	2	23
T	R4, TYC	3587	6.99	0.02	6	7

For these experiments, the following evaluation measures were considered: Probability Mass (PM), Sequence Error Rate (SER), Precision, and Recall [22]. PM is the total probability assigned by the model to the samples of the corpus (in the results, Probability Mass is normalized by the number of samples). SER is the percentage of samples recognized by the model with respect to the total number of samples (a sequence is recognized if the probability assigned by the model is greater than zero). Precision and recall measure the similarity between the best parse tree obtained by the Earley algorithm using SCFG and the real tree. Precision calculates the percentage of correct rules with respect to the total number of rules of the model parse tree. Recall calculates the percentage of correct rules with respect to the total number of rules of the real parse tree.

The first step was to carry out the HMM learning process of non-structured regions. For each region, we explored a set of models with different topologies: lineal, bakis, and left-right. We also used different numbers of states (from the minimum length to the maximum length of the training samples). Each model was trained using the Baum-Welch algorithm and the Viterbi algorithm [14]. The SER and PM measures from an independent test corpus were the criteria for selecting the best models. The best HMM selected were always estimated with the Baum-Welch algorithm. The best topologies for each region were: 1-state model for region 0; 6-state bakis for D-Loop; 7-state lineal for Anticodon loop; 4-state bakis for the variable loop; and 7-state lineal for TYC region [7]. Finally, the selected HMM were transformed to SCFG.

The second step was to learn the SCFG of the structured regions. This step requires two processes: to obtain the initial SCFG and to estimate these SCFG. The first process of learning the initial SCFG was broached by means of two methods: the Sakakibara-based algorithm and the Corpus-based algorithm.

The Sakakibara algorithm implementation used in the experiments is a general purpose software [16]. The only input of the algorithm is the training corpus composed by bracketed sequences as shown in Figure 3. The output is the SCFG made up of the rules and their associated probabilities.

The Corpus-based initialization algorithm was implemented as a tRNA sequence parser of the structured regions. The parser procedure advances along the sequence, and the positions of the nucleotides allow us to determine the consequent term of each rule. Therefore, the parser procedure generates the tree structure and increases the rule usage of each node of the tree simultaneously. When the parser procedure ended for each training sample, a simple estimation of the probabilities was calculated by counting rules.

The second process of re-estimation was carried out using the Inner-Outer algorithm. The implementation used in this work is a general purpose software [10]. The input is the training corpus, composed by bracketed sequences (see Figure 3), and the initial SCFG.

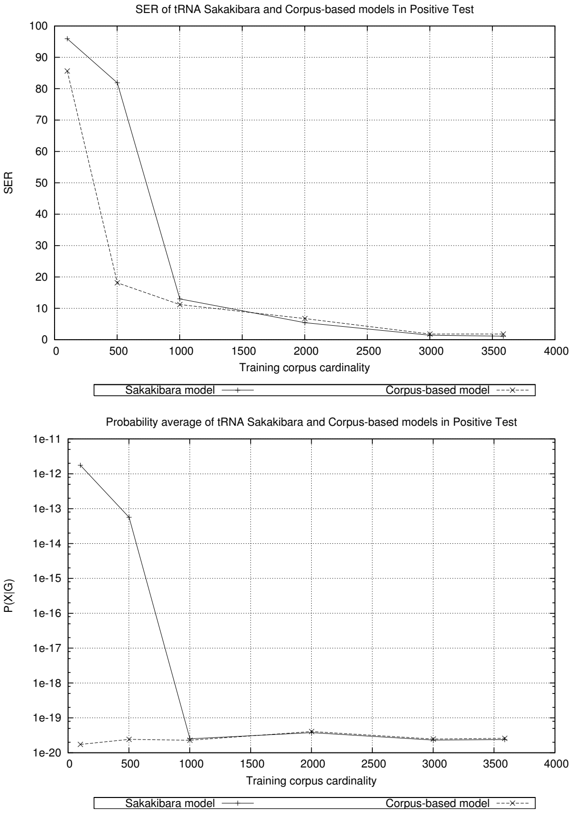

Figure 4: SER & PM for the two learning methods

In order to evaluate the proposal, three experiments are reported: 1) the behaviour of the models in the recognition of positive samples depending on the size of the training corpus; 2) the behaviour of the models in the recognition of negative samples and 3) the comparison of this proposal with those of other authors.

First, to evaluate the influence of the training size, six cumulative training subcorpora of 100, 500, 1000, 2000, 3000 and 3587 samples (the whole training corpus) were prepared.

The Sakakibara algorithm and Corpus-based algorithm were executed using the six subcorpora, and the models obtained were re-estimated using the bracketed Inner-Outer (IOb) algorithm.

Figure 4 shows the comparison of the SER and the PM of the initial models, using both the Sakakibara and the Corpus-based learning methods. Table 3 shows the same comparison with and without the re-estimation step (IOb or noIOb, respectively). The parameters used for measuring were the Size of the final Models (SofM), Senquence Error Rate (SER), Probability Mass (PM), Precision, and Recall.

Using a low number of training samples such as 100 or 500 (Figure 4 and Table 3), the Sakakibara algorithm did not achieve a good model whereas the Corpus-based algorithm began to converge when the training size were 500. Using a training size of 500, the Sakakibara algorithm obtained high sequence error rates (81.86%) in the positive test and non-coherent Probabily Mass. The difficult of that the Sakakibara algorithm had to converge using 500 samples is shown in the high number of rules of the models (1942 rules). In these models, computational problems did not allow the Precision and Recall parameters to be obtained. Fast convergence of the Corpus-based algorithm can be observed in the model trained with 500 samples. The straight guide Corpus-based algorithm obtained satisfactory results (18.14% in SER and 340 rules), whereas the Sakakibara algorithm did not condense rules with 500 training samples.

When the number of samples went over 1000 samples (figure 4 and Table 3), the Sakakibara algorithm condensed the rules to a specific model of tRNA molecules, and a similar probability mass was obtained for both strategies. Thus, both algorithms converged to similar rates, obtaining a very low SER (13%, 5,4%, 1.4%, 1.1%) for the Sakakibara, and a very low SER (11.18%, 6.7%, 1.8%, 1,8%) for Corpus-based using 1000, 2000, 3000 and full training corpus). A high precision rates was obtained for the Sakakibara method (99%) and for the Corpus-based method (98%)(Table 3). A high recall rates was obtained for the Sakakibara method (96%) and for the Corpus-based method (100%)(Table 3).

The Sakakibara initialization algorithm and the Corpus-based initialization algorithm both obtained good results using the full training corpus (3587 samples) for the tRNA modelling (1.13 % and 1.81 % in SER and more than 98 % in precision and recall for both methods).

The behaviour of the re-estimation algorithm was the convergence to the final model in only one iteration. The models obtained had a very small increase in the probability mass of the positive test samples and a small decrease in the sequence error rate (Table 3). The reestimated models had good behaviour in precision and recall rates similar to the initial models.

The second experiment was carried out to evaluate the behaviour of the learned models in accepting or rejecting negative test samples. The same procedure described in the previous experiment was reproduced and applied to the negative test corpus.

In this way, the generation of negative samples of similar lengths to tRNA were offered to the SCFG models using the Earley-based algorithm. As a result, no sequences of non-tRNA molecules were accepted by the Sakakibara nor by the Corpus-based initial models. Similar results were obtained for the re-estimated models. In summary, we can conclude that when the negative test was applied to the Sakakibara and the Corpus-based initial and reestimation models, 100 % SER was achieved in all cases. These results shows the specifity of the learned models. The topology of the SCFG combined with the simple HMM has the capacity to capture the well-known tRNA secondary structure.

The third experiment was carried out to compare the behaviour of our proposed method with previous works in the field. The tRNA-SE tool [12] was used to recognize both tRNA sequences (positive training) and non-tRNA sequences (negative training) corpora. The results of tRNA-SE were compared using the SER with the evaluation of our proposal carried out in the first and second experiments.

tRNAscan-SE model results using the positive test corpus achieved very good results. Most of the corpus was recognized, and only eight of the samples were not accepted. Thus, the SER of the tRNAscan-SE for the positive corpus was (0.6 %). The results of our proposal were 1.13 % SER using the Sakakibara algorithm and 1.81 % SER using the Corpus-based algorithm for the same corpus.

tRNAscan-SE model results using the negative test corpus also achieved very good results. Most of the corpus was not recognized by tRNAscan-SE, and only one negative sample (99.92% of SER) was accepted. The results of our proposal were 100 % of SER; that is, all the negative test samples were rejected.

Finally, the learning process carried out to obtain the tRNA models implies the analysis of large sets of samples, and, in some cases, iterative procedures, which could take a long computational time, therefore, the execution time of each step was measured on an Intel P4, 3GHz Debian-Linux Woody to compare the efficiency of the different algorithms.

When the Sakakibara algorithms was applied to the full-training corpus, it took five days to obtain the result for the full-training corpus. In contrast, the Corpus-based method took only a few seconds (less than 1 minute) to obtain the result for the full-training corpus. The reason for this difference is because the Sakakibara algorithm enters in

Table 3: Evaluation of the learning methods depending on the training corpus size.

Initialization	Re-estimation	TSize	SofM	SER (%)	PM	Precision (%)	Recall (%)
Sakakibara	noIOb	100	658	95.91	1.757177e-12	100	96.43
Sakakibara	IOb	100	658	95.91	1.757485e-12	100	96.43
Corpus-based	noIOb	100	266	85.63	1.734743e-20	100	100
Corpus-based	IOb	100	264	86.39	2.640565e-20	100	100
Sakakibara	noIOb	500	1942	81.85	5.655406e-14	-	-
Sakakibara	IOb	500	1942	81.85	5.657779e-14	-	-
Corpus-based	noIOb	500	340	18.14	2.418828e-20	98.29	100
Corpus-based	IOb	500	329	20.71	3.034496e-20	98.28	100
Sakakibara	noIOb	1000	476	13.00	2.506872e-20	99.98	96.41
Sakakibara	IOb	1000	476	13.00	2.508423e-20	99.98	96.41
Corpus-based	noIOb	1000	397	11.18	2.276320e-20	98.32	100
Corpus-based	IOb	1000	379	13.22	2.534100e-20	98.32	100
Sakakibara	noIOb	2000	542	5.442	3.766439e-20	99.94	96.37
Sakakibara	IOb	2000	542	5.442	3.767871e-20	100.00	96.43
Corpus-based	noIOb	2000	435	6.727	4.062833e-20	98.38	100
Corpus-based	IOb	2000	420	7.936	4.082250e-20	98.38	100
Sakakibara	noIOb	3000	645	1.436	2.308744e-20	99.92	96.35
Sakakibara	IOb	3000	645	1.436	2.305606e-20	99.92	96.35
Corpus-based	noIOb	3000	530	1.814	2.466287e-20	98.43	100
Corpus-based	IOb	3000	512	3.023	2.455825e-20	98.42	100
Sakakibara	noIOb	3587	662	1.133	2.385992e-20	99.92	96.35
Sakakibara	IOb	3587	662	1.133	2.382505e-20	99.92	96.35
Corpus-based	noIOb	3587	543	1.814	2.590248e-20	98.43	100
Corpus-based	IOb	3587	521	2.872	2.516052e-20	98.41	100

a iterative process of condensing rules, whereas the Corpus-based algorithm applies a simple frequency estimation calculated from counting rules. The reestimation bracketed version of the Inner-Outer algorithm reduced computational time to 40 % of the non-bracketed version.

4 Conclusions

The combination of SCFG and HMM allows us to characterize the secondary structure of tRNA and achieve good prediction rates. The SCFG are quite useful for representing the structured regions (the arms) of the tRNA molecule, and obtain fine models by using learning algorithms with structural information. HMM obtain good, simple models that avoid unnecessarily complex models in non-structured regions.

The Corpus-based method goes straight to the grammar model guided by the trees with labelled internal nodes. This strategy is better for reducing the size of the training corpus needed to get a good model. When a high number of samples is used, the Sakakibara-based method and the Corpus-based method achieve similar results.

The inner-outer reestimation algorithm obtains a small reduction in the number of rules and a small decrease in the sequence error rate of the SCFG models.

The Corpus-based grammar algorithm, with a fast reestimation step applied to small training corpus, might be a fast and reliable way to recognize the secondary structure of sequences with regions of palindromes. Thus, it should perform well with undiscovered ncRNA families, as miRNA and others.

The negative samples used in these experiments are synthetically extracted from real data but they do not incorporate a secondary structure that is close enough to tRNA. The testing of negative samples with different nucleotide pair structures could measure the specifity of the tRNA grammars. Further work will apply this method to classify tRNA molecules in groups that are annotated in the Steinberg database.

The proposed method described here automatically infers the secondary structure of tRNA molecules. This method can be applied to model the secondary structure of other ncRNA molecules. The combination of SCFG and HMM provides the necessary mechanism for modelling the long-term, and local relations in a unified model and provides efficient algorithms for automatic learning from annotated structural corpora.

5 ACKNOWLEDGEMENTS

We would like to thank Diego Linares and Joan Andreu Sánchez for answering all our questions about SCFG as well as Satoshi Sekine for his evaluation software. We would also like to thank the Ministerio de Sanidad y Consumo of Spain for the grants to the INBIOMED consortium.

6 *

References

[1] A V Aho and J D Ullman. The theory of parsing, translation and compiling. *Prestice-Hall*, 1972.

[2] E. Charniak. Tree-bank grammars. technical report. Technical report, Departament of Computer Science, Brown University, Providence, Rhode Island, January 1996.

[3] R Durbin, S Eddy, A Krogh, and G Mitchinson. Biological sequence analysis. *Cambridge University Press*, 1998.

[4] J Earley. An efficient contex-free parsing algorithm. *Communications of the ACM*, 8(6):451–455, 1970.

[5] Eddy and Durbin. Rna sequence analysis using covariance models. *Nucl. Acids Res.*, 22:2079–2088, 1994.

[6] Fichant and Burks. Identifying potential trna genes in genomic dna sequences. *J. Mol. Biol*, 220:659–671, 1991.

[7] J M Garcia-Gomez and J M Benedi. trna modelling by stochastic context-free grammars and hidden markov models. Technical Report DSIC-II/07/04, DSIC-UPV, 2004.

[8] Krogh and al. Hidden markov models in computational biology. *Journal of Molecular Biology*, 235:1501–1531, 1994.

[9] A Krogh, IS Mian, and D Haussler. A hidden markov model that finds genes in e. coli dna. *Nucleic Acids Research*, Vol 22(22):4768–4778, 1994.

[10] D Linares, JM Benedi, and JA Sanchez. A hybrid language model based on a combination of n-grams and stochastic context-free grammars. *ACM trans. on Asian Language Information Processing (TALIP)*, 3(2):113–127, 2004.

[11] D Linares, JM Benedi, and JA Sanchez. Learning of stochastic context-free grammars by means of estimation algorithms and initial treebank grammars. *In IbPRIA: Iberian Conference on Pattern Recognition and Image Analysis*, pages 403–410, June 2003.

[12] Lowe and Eddy. trnascan-se: A program for improved detection of transfer rna genes in genomic sequence. *Nucl. Acids Res.*, 25:955–964, 1997.

[13] Alexander V. Lukashin and Mark Borodovsky. Genemark.hmm: new solutions for gene finding. *Nucleic Acids Research,*, 26(4):1107,15, 1998.

[14] MacDonald and Zucchini. *Hidden Markov and Other Models for Discrete-valued Time Series.* Champan and Hall, 1997.

[15] William McClure. *tRNA Structure*. Deparment of Biological Sciences, Carnegie Mellon University, http://info.bio.cmu.edu/ Courses/ BiochemMols/ tRNA_Tour/ tRNAMain.htm, 2004.

[16] F. Nevado, J.A. Sanchez, and J.M. Benedi. Combination of estimation algorithms and grammatical inference techniques to learn stochastic context-free grammars. In A.L. Oliveira, editor, *Grammatical Inference Algorithms and Applications*, volume 1891 of *Lecture Notes in Computer Science*, pages 196–206. Springer-Verlag, 2000.

[17] Pavesi and at. Identification of new eukaryotic trna genes in genomic dna databases by a multistep weight matrix analysis of transcriptional control regions. *Nucl. Acids Res.*, 22:1247–1256, 1994.

[18] Y Sakakibara. Efficient learning of context-free grammars from positive structural examples. *Information and Computation*, 97:23–60, 1992.

[19] Y Sakakibara, M Brown, R Hughey, I Saira Mian, K Sjolander, R Underwood, and D Haussler. Stochastic context-free grammars for trna modeling. *Nucleic Acids Research*, 22:5112–5120, 1994.

[20] Y Sakakibara, M Brown, R Underwood, I Saira Mian, and D Haussler. Stochastic context-free grammars for modeling rna. *UCSC-CRL*, 1993.

[21] D Searls. The linguistic of dna. *American Scientist*, 80:579–591, 1992.

[22] Satoshi Sekine and Mike Collins. *evalb*, 1997.

[23] Mathias Sprinzl, C Horn, M Brown, A Ioudovitch, and S Steinberg. Compilation of trna sequences and sequences of trna genes. *Nucleic Acids Research*, 26(1):148–153, 1998.

[24] Mathias Sprinzl and Konstantin S Vassilenko. Compilation of trna sequences and sequences of trna genes. http://www.trna.uni-bayreuth.de, April 2003.

[25] A Stolcke. *Bayesian Learning of Probabilistic Language Models.* PhD thesis, University of California, Berkeley, CA, 1994.

[26] A Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–200, 1995.

[27] J Wuyts, P Rijk, Peer, T Winkelmans, and De Wachter. The european large subunit ribosomal rna database. *Nucleic Acids Res*, 29(1):175–177, 2001.

[28] S. Young. The htk hidden markov model toolkit: Design and philosophy, 1993.