IX Symposium on Pattern Recognition and Image Analysis. Vol.1, pp.171–176. Castellón, 2001.

RNA modeling by combining stochastic context-free grammars and n-gram models

Ismael Salvador and José-Miguel Benedí
Departamento de Sistemas Informáticos y Computación.
Universidad Politécnica de Valencia.
Camino de Vera, s/n, 46022 Valencia, SPAIN
e-mail: {isalva@iti.upv.es, jbenedi@dsic.upv.es}

Abstract

The RNA sentences present structured regions caused by palindrome pairs and non-structured regions where any global relations can be found. In this paper, we present a combination of stochastic context-free grammars (SCFG) and bigram models. The SCFGs are used to represent the long-term relations of the structured part, while the bigram models are used to capture the local relations of the non-structured part of RNA sentences. A stochastic version of the Sakakibara algorithm is used to learn the SCFGs. Finally, experiments to evaluate the behavior of this proposal were carried out.

Keywords: RNA, language modeling, grammatical inference, stochastic context-free grammars.

1 Introduction

Computer Science, Molecular Biology and other apparently distinct fields have come together in order to deal with the genome language. Currently, there is an increasing need to determine the sequence alignment, discriminate members of one family from non-members and discover new ones [1, 2]. DNA sends information to proteins by means of RNA, which is made up of four different nucleotides known as adenine (A), cytosine (C), guanine (G) and uracil (U). The linear arrangement that forms the nucleotides is called the *primary structure*. These can interact forming *secondary structure* elements such as helices, loops and bulges. The folding of an RNA sequence is mostly determined by the $A-U$ and $G-C$ Watson-Crick pairs as well as $G-U$ base pairs, which all constitute the well-known *biological palindromes* of the genome [1]. In this paper, we focus on the structural aspect rather than the biological meaning of the sequences, that is, we have to deal with strings having the A, C, G, U symbols. These

strings present structured regions caused by palindrome pairs and non-structured regions where any long term relation can be found.

Stochastic local models such as HMMs [3, 2], n-grams, etc, have been used in order to model RNA sequences. However, these models are not the most appropriate because palindrome paired positions are treated independently due to the restriction of the structure of these local models. Thus, a more powerful class of languages is needed to represent RNA. Searls, in [1], shows that a *context-free grammar* is able to represent the ramifications (long-term dependencies) caused by the palindrome pairing structure. Moreover, in order to incorporate stochastic information that permits an adequate modeling of the variability phenomena of the problem, Stochastic Context-Free Grammars (SCFGs) are finally considered as an appropriated model.

A stochastic version of the Sakakibara algorithm has been recently proposed [4, 5] and applied to a bracketed corpus in order to learn SCFGs. This algorithm works well with fully bracketed sentences, but it is not appropriate for those strings which have long non-bracketed regions. In this paper, we propose learning the structured (full bracketed) regions by means of the stochastic version of the Sakakibara algorithm [5], and the non-structured (non-bracketed) regions by means of the classical n-gram models. We also present the combination of the SCFGs and n-gram models.

Finally, in order to evaluate the behavior of this proposal, preliminary experiments with a corpus proposed in [8] were carried out and reasonable results were achieved.

2 Methodology

We define a CFG G, as a four-tuple (N, Σ, P, S), where N is a finite set of non-terminal symbols, Σ is a finite set of terminal symbols ($N \cap \Sigma = \emptyset$), P is a finite set of rules of the form $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (N \cup \Sigma)^+$ (we only consider grammars with no empty rules), and S is the initial symbol ($S \in N$). A CFG in Chomsky Normal Form (CNF) is a CFG in which the rules are of the form $A \rightarrow BC$ or $A \rightarrow a$ ($A, B, C \in N$ and $a \in \Sigma$).

A SCFG, G_s is a pair (G, p) where G is a CFG and p is a probability distribution over the grammar rules.

As pointed out in the previous section, SCFGs are adequate for generating the language of biological palindromes. For example, given the sequence:

`CAUCAGGGAAGAUCAACUCUUG`

the first two derivations should be $S \rightarrow CS_1G$, $S_1 \rightarrow AS_2U$ which represent the $G - C$, $A - U$ Watson-Crick base pairs.

The learning of SCFG

One of the principal requirements in the analysis of RNA is the database searching. Thus, we need to have models (languages) that can determine the likelihood of a sample. Inference from only positive data is much less powerful than inference from both positive and negative data. However, a good way to solve this problem is to add structural information to the positive samples. As described in [4], the Sakakibara algorithm infers the minimum reversible context-free grammar which is consistent with the input structural sample. The reversible context-free grammars are a normal form for general context-free grammars. Therefore, we are not restricted to a subclass of context-free languages. A G context-free grammar is said to be reversible if both the following hold:

1. It is invertible, that is, $A \to \alpha$ and $B \to \alpha$ in P implies $A = B$.
2. It is reset-free, that is, $A \to \alpha B \beta$ and $A \to \alpha C \beta$ in P implies $B = C$.

In order to obtain a SCFG, a stochastic version of the Sakakibara algorithm has been used [5]. Next, the SCFGs obtained by this algorithm are transformed to CNF in order to be used by the reestimation methods [7].

The input to the algorithm is a structural training sample, that is, a set of strings and a syntactic tree which is associated to each string. The syntactic tree is typically represented by brackets (see Figs. 1a, 1b). For example, the bracketed sequence corresponding to the sequence shown above would be:

```
[C[A[[U[C[A[G[G]]]G]A][A[G[A[U[CAAC]]U]C]U]]U]G]
```

which stands for *secondary structure*.

The Sakakibara algorithm is able to infer the rules that represent the base pairs. Nevertheless, non-aligned regions such as the `CAAC` group in the example below, make generalization (merging) difficult because they generate a lot of different subtrees. Thus, a great number of rules are inferred and a very high error rate is produced.

The proposal

Due to the variability of the non-aligned regions, the training corpus is relabeled in terms of labels which substitute these regions (see Fig. 1c). Next, a SCFG is learned from this relabeled corpus.

In order to model these non-aligned regions (see Figs. 1c, 1d), n-gram models were considered. The power of the n-gram model resides in: the strong relation in the natural language between local constraints, the consistence with the training data,

```
        <        D arm            > < Anticodon arm ><              Extra arm                 ><   T arm      >
              Base pairings                      Anticodon                                      Base pairings
(((((((  ((((                  )))) (((((  ===  )))))                                        (((((       ))))))))))))
GGGCGAAUAGUGUCa.G..C.-.G.G..G..AGCACACCAGACUUGCAAUCUGGU.......--A........G......-.......-.......GGAGGGUUCGAGUCCCUCUUUGUCCA

[[G[G[G[C[G[A[AUA[G[U[G[UCAGCGGGAG]C]A]C]A[C[C[A[G[ACUUGCAAU]C]U]G]G]UAGG[G[A[G[G[GUUCGAGUC]C]C]U]C]U]U]U]G]U]C]C]A]

[[G[G[G[C[G[A[E0[G[U[G[E1]C]A]C]A[C[C[A[G[E2]C]U]G]G]E3[G[A[G[G[E4]C]C]U]C]U]U]U]G]U]C]C]A]

AUA UCAGCGGGAG ACUUGCAAU UAGG GUUCGAGUC
```

Figure 1: a) Original, b) Bracketed, c) Relabeled sequence, d) Non-aligned subsequences.

the simple formulation and easy implementation. On the other hand, the n-gram model only uses the information provided by the last n words and so only makes use of local information. In particular, bigrams [6] are used, so the estimation of these models are well-known [6], and the calculation of the probability $P(w_i|w_{i-1})$ can be efficiently performed.

The bigram models associated to each non-aligned region are represented as Stochastic Finite State Automata (SFSA). Then, they are converted into a Stochastic Regular Grammar (SRG). Finally, the rules of the SCFG (structural model) and the SRG (variability model) are merged and transformed to CNF.

3 Experiments

In order to evaluate this proposal, we have tried to used data which were very similar to the data proposed in [8]. A total of 1400 sequences were used, 1200 for training and 200 for positive testing. The average length of the sentences after filtering was 80.5. These sequences (see Fig. 1a) are the aligned ones which were produced by the work of Sakakibara [8] [1].

In order to convert them to a more adequate format, a filtering process that eliminates the padding and puts the brackets within the strings was applied (see Fig. 1b).

As explained in the previous section, the Sakakibara algorithm did not work well because of the non-base paired regions. Preliminary experiments for 400 bracketed training sequences and a test of 200 sentences obtained a 90% sentence error rate (SER). The model obtained had 3900 rules.

[1] A file containing the sequences is available at `ftp://ftp.cse.ucsc.edu/pub/rna/trna.alignment`

We noticed that all the sequences had 5 non-bracketed regions, and these regions were substituted by new non-terminals labeled as EX. Table 1 shows the distribution of these regions. Following the examples, a relabeled sequence is shown in fig. 1c.

Set	# different substrings
E0	22
E1	420
E2	330
E3	353
E4	68

Table 1: Number of different substrings in each EX set.

As a consequence, a relabeled version of the training and test set were generated. A SCFG was learned using this new corpus. A 3.5% SER and a model of 525 rules were achieved on the labeled test set. This result demonstrates that the Sakakibara algorithm is appropriate for learning the language of *biological palindromes*.

Since bigrams are able to represent local relations, a *bigram* was inferred for each of the EX sets. Then, by substituting the EX terminals in the original SCFG by the corresponding rules obtained from the corresponding bigram model, a new SCFG was obtained. This final SCFG had a total size of 640 rules.

Now, we have a methodology that can model both parts of the RNA sequences. In addition, a set of negative samples is needed to test the real performance of the method. Thus, non-RNA sequences were obtained from the Ribosomial Database Project (RDP) in a way similar to the method used in [8] by cutting sequences into pieces of approximately the same lengths as RNA sequences.[2]

Table 2 reports the results with the combined model showing the SER and the accumulated sum of probabilities of the accepted sentences given the original test set of positive data and the new test set of negative data. These results show that the proposed methodology is adequate for learning the structure of RNA sequences, and consequently for discriminating between RNA and non-RNA sequences.

Set	SER	Probability sum
Positive	0%	2.16e-31
Negative	97.5%	1.60e-55

Table 2: Results with the combination of SCFG and bigrams.

[2]The file is publicly available at `ftp.cme.msu.edu/pub/RDP/LSU_rRNA/alignments/LSU.aln`

4 Conclusions

The combination of SCFGs and *bigrams* allow us to characterize the different parts of RNA sequences. The SCFGs are used to represent the long-term relations of the structured part, while the bigram models are used to capture the local relations of non-structured parts of RNA sentences. Preliminary experiments have shown that this methodology is able to discriminate between RNA and non-RNA sequences.

Further work will be directed towards obtaining new and bigger corpora in order to do a better estimation of the performance of this method.

References

[1] D. B. Searls, The linguistics of DNA, American Scientist, v. 80, 1992.

[2] R. Durbin et al., *Biological sequence Analysis*, Cambridge University Press, 1998.

[3] A.Krogh, M.Brown, I.S.Mian, K.Sjolander, D.Haussler. Hidden Markov Models in Computational Biology: Applications to Protein Modeling, Journal Molecular Biology, 235:1501–1531, February 1994.

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

[5] F. Nevado, J. A. Sánchez, J. M. Benedí, Combination of Estimation Algorithms and Grammatical Inference Techniques to Learn Stochastic Context-Free Grammars , *In 5th International Colloquium, ICGI 2000*, pp. 196-206, Lisbon, Portugal, September 2000.

[6] L. R. Bahl, F. Jelinek, R. L. Mercer, A maximum likelihood approach to continuous speech recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 2, pp 179-190, 1983.

[7] J.A. Sánchez and J.M. Benedí: Learning of Stochastic Context-Free Grammars by means of Estimation Algorithms. Proceedings of EUROSPEECH, pp.1799–1802, 1999.

[8] Y. Sakakibara, M. Brown, R. C. Underwood, I. S. Mian, D. Haussler, Stochastic context-free grammars for modeling RNA, Technical Report: UCSC-CRL-93-16, University of California, 1993.