DNA synthetic sequences generated by finite-context models*

Diogo Pratas, Armando J. Pinho, António J. R. Neves, and Carlos A. C. Bastos
Signal Processing Lab, DETI / IEETA
University of Aveiro, 3810–193 Aveiro, Portugal
pratas@ua.pt, ap@ua.pt, an@ua.pt, cbastos@ua.pt

Abstract

The development and implementation of computational models to represent DNA sequences is a great challenge. Models that rely exclusively on the Markov property, usually known as finite-context models, have been exploited in DNA data compression. In this paper, we present preliminary results regarding a study that aims at finding how well DNA synthetic sequences can be generated by finite-context models. These models provide probability estimates that depend on the recent past of the sequence in order to generate the next symbol. The experimental results show that synthetic sequences can be generated using these models, motivating further study.

1 Introduction

DNA generation is the process of sequence design, bringing into existence synthetic DNA data based on pseudo-random models. In this way, synthetic sequence generation has now an important role in better understanding some biological characteristics. Some methods have been proposed to date [2], [8], [7].

In this paper, we propose a method for DNA generation, relying on finite-context modelling, based exclusively on the Markov property [6]. It uses a model that captures the statistical information along the sequence in order to generate the next symbol. The preliminary results that we have obtained are promising, motivating further research efforts.

1.1 Finite-context models

Consider an information source that generates symbols (DNA bases), s, from the alphabet $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. Also, consider that the information source has already generated the sequence of n symbols $x^n = x_1 x_2 \ldots x_n,\ x_i \in \mathcal{A}$. A finite-context model (see Fig. 1) assigns probability estimates to the symbols of the alphabet, regarding the next outcome of the information source, according to a conditioning context computed over a finite and fixed number, $k > 0$, of the most recent past outcomes $c = x_{n-k+1} \ldots x_{n-1} x_n$ (order-k finite-context model) [1, 9, 10].

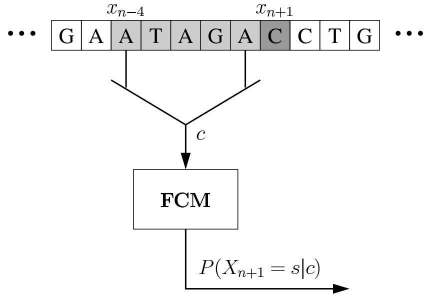

Figure 1. Example of a finite-context model: the probability of the next outcome, X_{n+1}, is conditioned by the last k outcomes. In this example, $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and $k = 5$.

The probability estimates, $P(X_{n+1} = s|c), \forall_{s \in \mathcal{A}}$, are usually calculated using symbol counts that are accumulated while the sequence is processed, which makes them dependent not only of the past k symbols, but also of n. In other words, these probability estimates are generally time varying.

In practice, the probability that the next outcome, X_{n+1}, is s, where $s \in \mathcal{A}$, is obtained using the estimator

$$P(X_{n+1} = s|c) = \frac{n_s^c + \alpha}{\sum\limits_{a \in \mathcal{A}} n_a^c + 4\alpha}, \qquad (1)$$

where n_s^c represents the number of times that, in the past, the information source generated symbol s having c as the conditioning context. Parameter α controls how much probability is assigned to unseen (but possible) events, and plays

*This work was supported in part by the grant with the COMPETE reference FCOMP-01-0124-FEDER-010099 (FCT, Fundação para a Ciência e Tecnologia, reference PTDC/EIA-EIA/103099/2008).

a key role in the case of high order models. When k is large, the number of conditioning states, 4^k, is high, which implies that statistics have to be estimated using only a few observations. Note that this estimator is the Laplace estimator when $\alpha = 1$ [5] and the Jeffreys [3] / Krichevsky-Trofimov estimator [4] when $\alpha = 1/2$.

2 Experimental results and conclusions

For the evaluation of the generation method we used the release of April 14th, 2003 of the Human genome (`ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/April_14_2003`), specially chromosome 2 and Y. From chromosome 2, it was chosen a sequence of one million symbols. The sequence was compressed using a finite-context model [6], for model orders from one to 14, and the final entropy value was obtained in each case. Then, using the models created from the original sequence, we generated the synthetic sequence, with the same size as the original and $\alpha = 1/10$. Then, the synthetic sequence was compressed using the same method as for the original sequence, and the final entropy value was obtained. Fig. 2 shows those entropy values. As can be seen, for low orders the two curves coincide, but, as expected, for higher orders the synthetic sequence is better compressed.

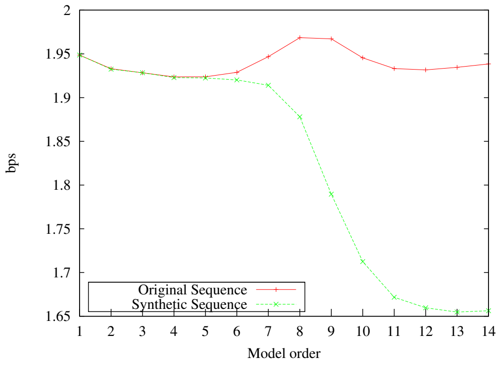

Figure 2. Original and the corresponding synthetic sequences compressed.

We have used the same process as described above, in order to test chromosome Y. As we know from previous studies, the Y chromosome is highly repetitive, a property that produces high compression ratios. In Fig 3, it can be seen that for high orders and $\alpha = 1/10$ the synthetic sequences are better compressed. However, for $\alpha = 1$ the entropy approaches the 2 bits per symbol (bps) limit.

Finite-context modelling has been used for a long time for DNA compression. In this paper, we have shown that finite-context modelling can also be used for sequence generation. It is known that DNA is better represented by multiple finite-context models [6]. However, the current generator only allows a single finite-context model to be used for generating a certain sequence. Therefore, in the future, we intend to improve our generator by exploring this characteristic.

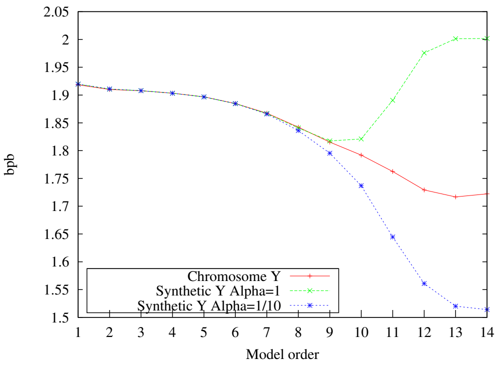

Figure 3. Chromosome Y and the corresponding synthetic sequence compressed with different α values.

References

[1] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text compression*. Prentice Hall, 1990.

[2] U. Feldkamp, S. Saghafi, W. Banzhaf, and H. Rauhe. DNASequenceGenerator: a program for the construction of DNA sequences. In *DNA Computing: 7th Int. Workshop on DNA-Based Computers, DNA7*, volume 2340 of *LNCS*, pages 23–32, 2001.

[3] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proc. of the Royal Society (London) A*, 186:453–461, 1946.

[4] R. E. Krichevsky and V. K. Trofimov. The performance of universal encoding. *IEEE Trans. on Information Theory*, 27(2):199–207, Mar. 1981.

[5] P. S. Laplace. *Essai philosophique sur les probabilités (A philosophical essay on probabilities)*. John Wiley & Sons, New York, 1814. Translated from the sixth French edition by F. W. Truscott and F. L. Emory, 1902.

[6] A. J. Pinho, A. J. R. Neves, D. A. Martins, C. A. C. Bastos, and P. J. S. G. Ferreira. Finite-context models for DNA coding. In S. Miron, editor, *Signal Processing*, pages 117–130. INTECH, Mar. 2010.

[7] Y. Ponty, M. Termier, and A. Denise. GenRGenS: software for generating random genomic sequences and structures. *Bioinformatics*, 22(12):1534–1535, 2006.

[8] E. Rouchka and C. Hardin. rMotifGen: random motif generator for DNA and protein sequences. *BMC Bioinformatics*, 8(1):292, 2007.

[9] D. Salomon. *Data compression - The complete reference*. Springer, 4th edition, 2007.

[10] K. Sayood. *Introduction to data compression*. Morgan Kaufmann, 3rd edition, 2006.