BACTERIA DNA SEQUENCE COMPRESSION USING A MIXTURE OF FINITE-CONTEXT MODELS

Armando J. Pinho, Diogo Pratas, Paulo J. S. G. Ferreira

Signal Processing Lab, IEETA / DETI
University of Aveiro, 3810–193 Aveiro, Portugal
{ap,pratas,pjf}@ua.pt

ABSTRACT

The ability of finite-context models for compressing DNA sequences has been demonstrated on some recent works. In this paper, we further explore this line, proposing a compression method based on eight finite-context models, with orders from two to sixteen, whose probabilities are averaged using weights calculated through a recursive procedure. The method was tested on a total of 2,338 sequences belonging to bacterial genomes, with sizes ranging from 1,286 to 13,033,779 bases, showing better compression results than the state-of-the-art XM DNA coding algorithm and also faster operation.

Index Terms— DNA sequences, finite-context models, data compression

1. INTRODUCTION

Almost twenty years ago, Grumbach and Tahi [1] proposed the first algorithm dedicated to DNA coding, *Biocompress*. Since this work, several other contributions have been made in the area of DNA data compression (see, for example, [2, 3, 4, 5, 6, 7, 8, 9]). Most of these works explore the non-stationary nature of the DNA sequence data, typically using at least two encoding methods, one based on a Lempel-Ziv-like substitutional procedure [10] and another based on a low-order context-based arithmetic coding.

According to the substitutional paradigm, repeated regions of the DNA sequence are represented by a pointer to a past occurrence of the repetition and by the length of the repeating sequence. Both exact and approximate repetitions have been explored, as well as their inverted complements. In the case of approximate repetitions, it is also required to indicate where and both the sequences differ.

The substitutional approach is usually the main encoding method, with the low-order finite-context model assuming the role of a fall-back, secondary choice. When the substitutional method is unable to provide satisfactory performance, the corresponding region of the DNA sequence is represented by a low-order finite-context model. This scheme for representing DNA data has been significantly improved by Tabus and by Korodi et al., based on the normalized maximum likelihood (NML) algorithm [11, 6, 8], and by Cao *et al.* [9], using the so-called expert model (XM).

The most recent version of the NML-based approach [8] is an evolution of the normalized maximum likelihood model introduced in [11] and improved in [6]. This new version, NML-1, aims at finding the best regressor block, i.e., an approximate repetition, using first-order dependencies (these dependencies were not considered in the previous approaches).

The other recent method, XM [9], relies on a mixture of experts for providing symbol by symbol probability estimates, which are then used for driving an arithmetic encoder. The algorithm is based on order-2 Markov experts, on order-1 context Markov models (models that use statistical information only of a recent past), and on copy expert (these consider the next symbol as part of a copied region from a particular offset). The probabilities provided by the set of experts are combined using Bayesian averaging and used to drive the arithmetic encoder.

Recent work has shown that the ability of finite-context modeling to represent DNA data sequences seems to go beyond a simple secondary role [12, 13]. In [13] we have addressed the problem of DNA sequence compression using two finite-context models competing for encoding the data on a block basis. The idea was to use a low-order model for modeling high information content regions and a high-order model for those region having low entropy. The method described in [13] is forward-adaptive, therefore requiring side information, in this case for indicating which of the two models encodes the block.

In this paper, we explore, on one hand, a backward-adaptive approach and, on the other hand, the use of more than two simultaneous finite-context models. We tested the codec on the complete set of bacteria DNA sequences available on the NCBI website, and we compared its performance with the state-of-the-art XM encoder. The results show that the proposed approach not only provides better compression, but it is also considerably faster.

2. THE COMPRESSION METHOD

2.1. Finite-context models

For DNA sequences, a finite-context model assigns probability estimates to the symbols of the alphabet $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, regarding the next outcome, according to a conditioning context computed over a finite and fixed number, $k > 0$, of the most recent past outcomes $x_{n-k+1..n} = x_{n-k+1} \dots x_n$ (order-k finite-context model) [14].

The probability estimates $P(x_{n+1}|x_{n-k+1..n})$ are calculated using symbol counts that are accumulated while the sequence is processed, making them dependent not only on the past k symbols, but also on n. We use the estimator

$$P(s|x_{n-k+1..n}) = \frac{C(s|x_{n-k+1..n}) + \alpha}{C(x_{n-k+1..n}) + 4\alpha}, \tag{1}$$

where $C(s|x_{n-k+1..n})$ represents the number of times that, in the past, symbol s was found having $x_{n-k+1..n}$ as the conditioning context and where

$$C(x_{n-k+1..n}) = \sum_{a \in \mathcal{A}} C(a|x_{n-k+1..n}) \tag{2}$$

is the total number of events that has occurred so far in association with context $x_{n-k+1..n}$. Parameter α allows balancing between the maximum likelihood estimator and an uniform distribution. Note that when the total number of events, n, is large, the estimator behaves as a maximum likelihood estimator. For $\alpha = 1$, (1) is the well-known Laplace estimator.

The per symbol information content average provided by the finite-context model of order-k, after having processed n symbols, is given by

$$H_{k,n} = -\frac{1}{n} \sum_{i=0}^{n-1} \log_2 P(x_{i+1}|x_{i-k+1..i}) \quad \text{bpb}, \tag{3}$$

where "bpb" stands for bits per base. When using several models simultaneously, the $H_{k,n}$ can be viewed as measures of the performance of those models until that instant. Therefore, the probability estimate can be given by a weighted average of the probabilities provided by each model, according to

$$P(x_{n+1}) = \sum_k P(x_{n+1}|x_{n-k+1..n})\ w_{k,n}, \tag{4}$$

where $w_{k,n}$ denotes the weight assigned to model k and

$$\sum_k w_{k,n} = 1. \tag{5}$$

2.2. The mixture weights

For stationary sources, we could compute weights such that

$$w_{k,n} = P(k|x_{1..n}), \tag{6}$$

i.e., according to the probability that model k has generated the sequence until that point. In that case, we would get

$$w_{k,n} = P(k|x_{1..n}) \propto P(x_{1..n}|k)P(k), \tag{7}$$

where $P(x_{1..n}|k)$ denotes the likelihood of sequence $x_{1..n}$ being generated by model k and $P(k)$ denotes the prior probability of model k. Assuming

$$P(k) = \frac{1}{K}, \tag{8}$$

where K denotes the number of models, we also obtain

$$w_{k,n} \propto P(x_{1..n}|k). \tag{9}$$

Calculating the logarithm we get

$$\log_2 P(x_{1..n}|k) = \log_2 \prod_{i=1}^{n} P(x_i|k, x_{1..i-1}) = \tag{10a}$$

$$= \sum_{i=1}^{n} \log_2 P(x_i|k, x_{1..i-1}), \tag{10b}$$

which corresponds to the code length that would be required by model k for representing the sequence $x_{1..n}$. It is, therefore, the accumulated measure of the performance of model k until instant n. However, since the DNA sequences are not stationary, a good performance of a model in a certain region of the sequence might not be attained in other regions. Hence, the performance of the models have to be measured in the recent past of the sequence, for example over a window of appropriate size, or be equipped with a mechanism of progressive forgetting of past measures. We opted for the latter possibility, using the recursive relation

$$\sum_{i=1}^{n} \log_2 P(x_i|k, x_{1..i-1}) = \tag{11a}$$

$$= \gamma \sum_{i=1}^{n-1} \log_2 P(x_i|k, x_{1..i-1}) + \log_2 P(x_n|k, x_{1..n-1}). \tag{11b}$$

As can be easily verified, this relation corresponds to a first-order recursive filter that, for $\gamma \in [0, 1)$, has a low-pass characteristic and an exponentially decaying impulse response. The effect is to reduce the importance of the performance attained by the model on parts of the sequence far into the past and, therefore, to pay more attention to its performance in the recent past. Defining

$$p_{k,n} = \prod_{i=1}^{n} P(x_i|k, x_{1..i-1}), \tag{12}$$

and removing the logarithms, we can rewrite (11) as

$$p_{k,n} = p_{k,n-1}^{\gamma} P(x_n|k, x_{1..n-1}) \tag{13}$$

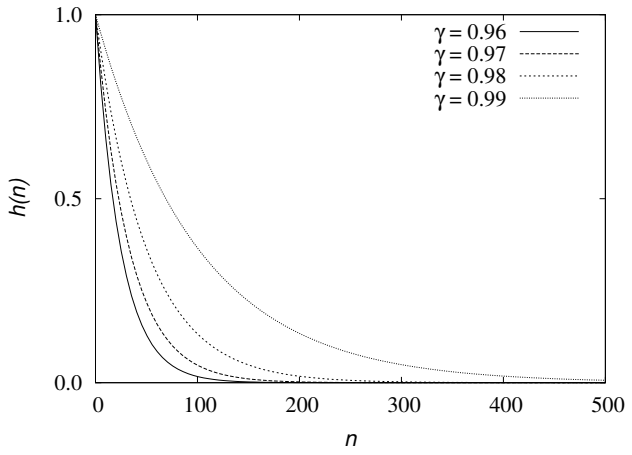

Fig. 1. Impulse response of the system described by (11) for several values of the parameter γ.

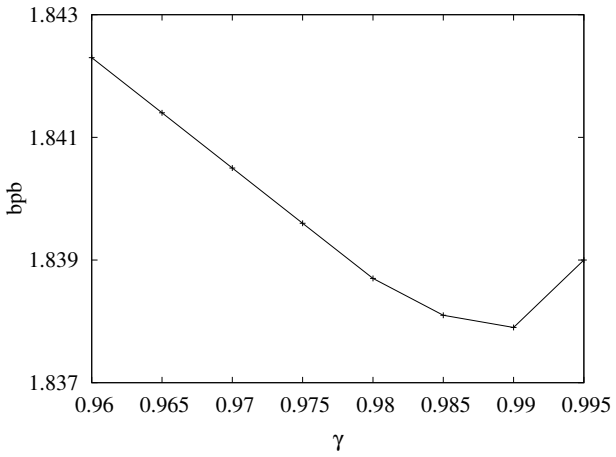

Fig. 2. Variation in the average number of bits per base as a function of parameter γ. The evaluation was performed using all available sequences.

and, finally,

$$w_{k,n} = \frac{p_{k,n}}{\sum_k p_{k,n}}. \quad (14)$$

If we calculate the impulse response of the system described by (11), we obtain

$$h(n) = \gamma^n, \quad n \geq 0, \quad (15)$$

and, plotting $h(n)$ for several values of the parameter γ, we can observe how the forgetting mechanism operates as a function of γ (see Fig. 1 for some examples).

3. EXPERIMENTAL RESULTS

For the experiments, we used the bacteria DNA sequences collected from the National Center for Biotechnology Information (NCBI) directory `ftp://ftp.ncbi.nih.gov/genomes/Bacteria/` on the 26th January 2011. This set contains 2,338 sequences, with sizes ranging from 1,286 to 13,033,779 bases and totaling 4,531,483,412 bases.

For the results presented in this paper we used a setup composed of eight finite-context models with orders $k = 2, 4, 6, 8, 10, 12, 14, 16$. The probabilities associated to the finite-context models were estimated using (1), with $\alpha = 1$ (corresponding to Laplace's estimator) for model orders $k = 2, 4, 6, 8, 10, 12$ and with $\alpha = 0.05$ for model orders $k = 14, 16$. The value of α is not too much important for low-order models, but it is crucial in high-order ones. In the latter case, the number of times that a given context occurs is generally small, rendering the estimation of the probability strongly dependent of α. From our experience, we found that using $\alpha = 0.05$ would provide, globally, good results. However, the performance of the estimator is robust with respect to small variations of α. Moreover, as suggested in [12], the finite-context counters corresponding to the inverted complements are also updated. For comparison, we used the XM encoder [9], currently considered the state-of-the-art method for DNA compression, limiting the number of experts to 500.

		XM-500		FCM-Mx	
Sequence	Size	Time	Bpb	Time	Bpb
NC_013929	10,148,695	512	1.786	71	1.754
NC_014318	10,236,715	507	1.789	73	1.739
NC_013595	10,341,314	527	1.796	73	1.759
NC_013131	10,467,782	509	1.817	75	1.779
NC_010162	13,033,779	575	1.755	92	1.743

Table 1. Detailed results, for the XM compression method using 500 experts (XM-500) and for the mixture of finite-context models (FCM-Mx), considering only the sequences with size equal or larger than 10,000,000 bases. Time is in seconds.

Figure 2 displays the variation of the compression efficiency of the proposed method as a function of the parameter γ. As can be seen, for the test set used, the best value is $\gamma = 0.99$. Moreover, the variation in the compression efficiency between using $\gamma = 0.96$ and $\gamma = 0.99$ is about 0.2%, showing a comfortable insensitivity to a possible mistuning of γ.

Table 1 presents the individual compression results obtained on the (five) sequences with 10,000,000 or more bases, as well as the time required to compress them. Comparison results obtained with XM using 500 experts are also included. Table 2 shows average compression measured on five classes of sequences: those having 10,000,000 or more bases, 5,000,000 or more bases, 1,000,000 or more bases, 500,000 or more bases, and the totality of the sequences. As in Table 1, we also present in this table the performance attained

Size	Sequences	Bases	XM-500 Time	XM-500 Bpb	FCM-Mx Time	FCM-Mx Bpb
$\geq$ 10,000,000	5	54,228,285	2,630	1.787	384	1.754
$\geq$ 5,000,000	217	1,313,760,062	33,442	1.841	9,294	1.821
$\geq$ 1,000,000	1,294	4,370,436,333	73,144	1.855	30,891	1.838
$\geq$ 500,000	1,386	4,439,121,955	73,614	1.854	31,400	1.837
$\geq$ 0	2,338	4,531,483,412	80,860	1.854	39,535	1.838

Table 2. Number of sequences, according to five size classes, total number of bases in each class, results for the XM compression method using 500 experts (XM-500) and for the mixture of finite-context models (FCM-Mx) proposed in this paper. Time is in seconds and corresponds to the total encoding time required for each of the five classes.

by the XM method.

4. CONCLUSION

The results presented in this paper strengthen the conclusion that has been drawn in other works [12, 13], i.e., that finite-context modeling plays an important role in the representation of DNA sequences. In the particular case addressed here, applied to bacterial genomes, we attained results that are better not only in terms of compression ratio, but also concerning the computing time, compared to the state-of-the-art XM.

The proposed method relies on a mixture of probabilities estimated by a set of cooperating finite-context models, where the mixture weights are related to the performance of the models and are obtained using a recursive procedure that provides a forgetting mechanism essential to cope with the non-stationary nature of the DNA sequences.

5. ACKNOWLEDGMENT

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).

6. REFERENCES

[1] S. Grumbach and F. Tahi, "Compression of DNA sequences," in *Proc. of the Data Compression Conf., DCC-93*, Snowbird, Utah, 1993, pp. 340–350.

[2] E. Rivals, J.-P. Delahaye, M. Dauchet, and O. Delgrange, "A guaranteed compression scheme for repetitive DNA sequences," in *Proc. of the Data Compression Conf., DCC-96*, Snowbird, Utah, 1996, p. 453.

[3] D. Loewenstern and P. N. Yianilos, "Significantly lower entropy estimates for natural DNA sequences," in *Proc. of the Data Compression Conf., DCC-97*, Snowbird, Utah, Mar. 1997, pp. 151–160.

[4] X. Chen, S. Kwong, and M. Li, "A compression algorithm for DNA sequences," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, pp. 61–66, 2001.

[5] G. Manzini and M. Rastero, "A simple and fast DNA compressor," *Software—Practice and Experience*, vol. 34, pp. 1397–1411, 2004.

[6] G. Korodi and I. Tabus, "An efficient normalized maximum likelihood algorithm for DNA sequence compression," *ACM Trans. on Information Systems*, vol. 23, no. 1, pp. 3–34, Jan. 2005.

[7] B. Behzadi and F. Le Fessant, "DNA compression challenge revisited," in *Combinatorial Pattern Matching: Proc. of CPM-2005*, Jeju Island, Korea, June 2005, LNCS, Springer-Verlag.

[8] G. Korodi and I. Tabus, "Normalized maximum likelihood model of order-1 for the compression of DNA sequences," in *Proc. of the Data Compression Conf., DCC-2007*, Snowbird, Utah, 2007.

[9] M. D. Cao, T. I. Dix, L. Allison, and C. Mears, "A simple statistical algorithm for biological sequence compression," in *Proc. of the Data Compression Conf., DCC-2007*, Snowbird, Utah, 2007.

[10] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. on Information Theory*, vol. 23, pp. 337–343, 1977.

[11] I. Tabus, G. Korodi, and J. Rissanen, "DNA sequence compression using the normalized maximum likelihood model for discrete regression," in *Proc. of the Data Compression Conf., DCC-2003*, Snowbird, Utah, 2003, pp. 253–262.

[12] A. J. Pinho, A. J. R. Neves, and P. J. S. G. Ferreira, "Inverted-repeats-aware finite-context models for DNA coding," in *EUSIPCO-2008*, Lausanne, Switzerland, Aug. 2008.

[13] A. J. Pinho, A. J. R. Neves, C. A. C. Bastos, and P. J. S. G. Ferreira, "DNA coding using finite-context models and arithmetic coding," in *ICASSP-2009*, Taipei, Taiwan, Apr. 2009.

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