Compression of DNA microarrays using a mixture of finite-context models

Luís M. O. Matos
luismatos@ua.pt

António J. R. Neves
an@ua.pt

Armando J. Pinho
ap@ua.pt

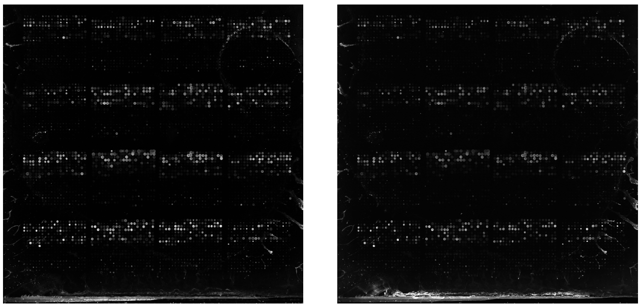
Signal Processing Laboratory
IEETA/DETI
University of Aveiro,
3810-193 Aveiro, Portugal

Abstract

DNA microarray images are experiments that allow the identification of the function and regulation of a large number of genes in a single experiment. These micrarray experiments consist of a pair (green and red channel) of 16 bits per pixel grayscale images. This paper addressed a lossless method to compress these kind of images, using a mixture of finite-context models and arithmetic coding. We use a 3D context configuration, expectation-based bitplane coding and also a typical image context template in the mixture. We conclude that using a mixture of finite-context models we are able to improve the compression results.

1 Introduction

We can use DNA microarray technology to study and monitoring gene function across a large number of genes, and even entire genomes. The raw data resulting from a microarray experiment consist of a pair of 16 bits per pixel grayscale images (see Figure 1). These images usually require tens of megabytes to be stored or transmitted without any compression. Due to this fact, and the need for long-term storage and efficient transmission, a lossless compression method with progressive decoding capabilities is an important challenge. In the literature we can find several proposals for lossless compression of microarray images [1, 4, 5, 6, 7]. In [3] we can find a review of DNA microarray image compression, where the authors describe the most relevant approaches published in the literature.



(a) Green channel

(b) Red channel

Figure 1: Example of a pair of images (1041 × 1044 pixels) that results from a microarray experiment.

2 Method

The proposed method is based on a mixture of finite-context models, where several models are used to estimate the probability of each symbol. Each model generates a probability estimate and the final probability is a weighted sum of the probabilities estimated by all models used.

2.1 Finite-context models

A finite-context model assigns probabilities to the symbols of an alphabet $\mathcal{A}$, according to a conditioning context. The probability estimates $P(X_{n+1} = s|c^t)$ are calculated using symbol counts that are accumulated while each bitplane of the microarray image is compressed. We use the estimator

$$P(X_{n+1} = s|c^t) = \frac{C(s|c^t) + \alpha}{\sum_{a \in \mathcal{A}} C(a|c^t) + |\mathcal{A}|\alpha}. \tag{1}$$

Parameter α allows balancing between the maximum likelihood estimator and an uniform distribution. 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

$$H_{k,n} = -\frac{1}{n} \sum_{i=0}^{n-1} \log_2 P(X_{i+1} = s|c^t) \text{ bps}, \tag{2}$$

where "bps" stands for bits per symbol.

2.2 Expectation-based bitplane Coding (EBC)

In 2010, Chen *et al.* [2] proposed a lossless compression algorithm that uses EBC. EBC is a strategy that is useful in some bitplanes where the pixel values are not so meaningful as they seem. Sometimes, the neighboring bits are only refining bits that fine-tune the value of a particular pixel. This model can be used in our approach in microarray images. Suppose that $p = \{p_{15}, p_{14}, \ldots, p_2, p_1, p_0\}$, where p_i is the value of the pixel at bitplane i. When we are encoding the n$^{\text{th}}$ bitplane, if the pixel p is already encoded at the current bitplane, its expectation value is formulated as

$$E(x) = \sum_{i=n}^{15} 2^i + (2^{n-1} - 1). \tag{3}$$

On the other hand, if the pixel p has not been encoded at current bitplane, its expectation value is defined as:

$$E(x) = \sum_{i=n}^{15} 2^i + (2^n - 1). \tag{4}$$

During the compression process, we use this expectation values instead of the real bit values. In case of using the EBC, if the expected value of the context is lower than the expected value of the current pixel, the context bit used is 0, otherwise 1 is used. The variable size template used is described in Figure 2 and as we can see, there are two future pixels (pixels 4 and 3) that are used in the context template. The pixel denoted by 0 represents the current pixel being compressed. Using this approach, we can select future pixels for the context template, which is more efficient for the least significant bitplanes, where the neighboring bits of each bitplane are merely refining bits and they only fine-tune the final value of that particular pixel.

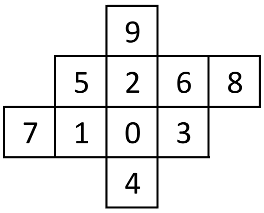


Figure 2: Context template used in EBC.

2.3 Proposed approach

As already said, the proposed method is based in a mixture of finite-context models. The algorithm processes the microarray images bitplane by bitplane using several models. We used the 3D finite-context model (see Figure 3) proposed in [6], a typical context template (see Figure 4) and also the EBC model presented in the previous subsection. The goal of our approach is to compute a probability estimate of several models and to combine them into a single probability that is used to compress each symbol. Each model contributes to the final probability estimate of the next outcome symbol. Therefore, we can compute the probability estimate using a weighted average of the probabilities provided by each model, according to

$$P(x_{n+1}) = \sum_k P(X_{n+1} = s|c^t)\, w_{k,n}, \qquad (5)$$

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

$$\sum_k w_{k,n} = 1. \qquad (6)$$

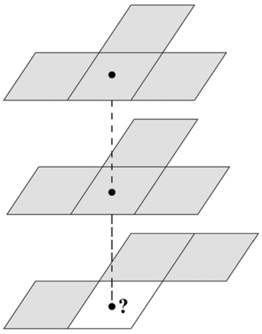


Figure 3: 3D context configuration proposed in [6].

We use a typical context template configuration similar to the template presented in Figure 4. The proposed algorithm supports several template configurations at the same time.

	10	6	9	
8	4	2	3	7
5	1	X		

Figure 4: An example of context template configuration used in our approach.

3 Results and Conclusion

In order to be able to compare our results with the results presented in [6], the microarray images used were collected from three publicly available sources. 1) 32 images that we refer to as the APO_AI set; 2) 14 images forming the ISREC set; 3) three images previously used in MicroZip. The image sizes ranges from 1000 × 1000 to 5496 × 1956 pixels and all of them have 16 bits per pixel. The average results presented take into account the different sizes of the images, i.e., they correspond to the total number of bits divided by the total number of image pixels.

In Table 1, we have the compression results of 3 standards (JPEG2000, JPEG-LS and JBIG), the algorithm proposed in [6] and our approach (rows “Mix10” and “MixEBC”). The “Mix10” corresponds to a mixture between the 3D context configuration presented in Neves [6] and the context template configuration presented in Figure 4. As we can see, there are small improvements in all datasets for the “Mix10”. On the other hand, the “MixEBC” which correspond to a mixture between the same 3D context configuration of “Mix10” and EBC, generates worst compression results. Table 2 presents another interesting metric, which is the number of bits per pixel that would required if we could select always the best model to encode each symbol (but without considering the side information indicating the model used). Therefore, this lower bound values describes how good is the quality of the model mixing procedure. The use of EBC (row “MixEBC”) does not provide good compression results in the mixture, compared to the use of a typical image context template similar to the one presented in Figure 4. However, according to Table 2, the lower bound value obtained is smaller than that for “Mix10”. We can conclude that, despite using the EBC model in the mixture, it does not provide better results, although in a competitive approach it could be more efficient than the “Mix10” approach.

Dataset Method	APO_AI	ISREC	Microzip	Average
JPEG2000	11.063	11.366	9.515	10.653
JBIG	11.851	10.925	9.297	10.393
JPEG-LS	10.608	11.145	8.974	10.218
Micro3DEnc [6]	10.314	10.199	8.667	9.617
Mix10	10.302	10.194	8.662	9.610
MixEBC	10.316	10.214	8.674	9.625

Table 1: Compression results in bits per pixel for the proposed method (“Mix10” and “MixEBC”), and the method presented in [6]. We included also the results for 3 image compression standards (JPEG2000, JPEG-LS and JBIG).

Dataset Method	APO_AI	ISREC	Microzip	Average
Mix10	9.571	9.564	7.954	8.915
MixEBC	9.460	9.536	7.848	8.828

Table 2: Lower bound of each proposed method. These values represent the number of bits per pixel that would be required if we could select the best model to encode each pixel (but without considering the side information indicating the model used).

The proposed method, based on a mixture of finite context-models, as we can see, provides small improvements compared to the method [6]. As future work, it will be interesting to explore other models than the EBC in the mixture. Also, we could use some pre-processing techniques in the microarray images in order to improve the compression results.

4 Funding

Supported by the European Fund for Regional Development (FEDER) through the Operational Program Competitiveness Factors (COMPETE) and by the Portuguese Foundation for Science and Technology (FCT) in the context of the project FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011).

References

[1] S. Battiato and F. Rundo. A bio-inspired CNN with re-indexing engine for lossless DNA microarray compression and segmentation. In *Proc. of the IEEE Int. Conf. on Image Processing, ICIP-2009*, volume 1-6, pages 1737–1740, Cairo, Egypt, November 2009.

[2] M. Chen, P. Franti, and M. Xu. Lossless bit-plane compression of images with context tree modeling. In *Int. Conf. on Green Circuits and Systems (ICGCS)*, pages 605–610, June 2010.

[3] M. Hernandez-Cabronero, I. Blanes, J. Serra-Sagrista, and M.W. Marcellin. A review of DNA microarray image compression. In *Proc. of Int. Conf. on Data Compression, Communication and Processing, CCP-2011*, pages 139–147, Palinuro, Italy, June 2011.

[4] R. Jörnsten, Y. Vardi, and C.-H. Zhang. On the bitplane compression of microarray images. In Y. Dodge, editor, *Proc. of the 4th Int. Conf. on Statistical Data Analysis on the L1-norm and Related Methods*, Neuchâtel, Switzerland, August 2002.

[5] A. Neekabadi, S. Samavi, S. A. Razavi, N. Karimi, and S. Shirani. Lossless microarray image compression using region based predictors. In *Proc. of the IEEE Int. Conf. on Image Processing, ICIP-2007*, volume 2, pages 349–352, San Antonio, Texas, USA, September 2007.

[6] A. J. R. Neves and A. J. Pinho. Lossless compression of microarray images using image-dependent finite-context models. *IEEE Trans. on Medical Imaging*, 28(2):194–201, February 2009.

[7] Y. Zhang, R. Parthe, and D. Adjeroh. Lossless compression of DNA microarray images. In *Proc. of the IEEE Computational Systems Bioinformatics Conference, CSB-2005*, Stanford, CA, August 2005.